

Dynamic Response of a Pulse-Heated, Thick- Walled Hollow Sphere: Validation of Code Numerics

R. E. Canaan

January 19, 2000

U.S. Department of Energy

Lawrence
Livermore
National
Laboratory

DISCLAIMER

This document was prepared as an account of work sponsored by an agency of the United States Government. Neither the United States Government nor the University of California nor any of their employees, makes any warranty, express or implied, or assumes any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise, does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States Government or the University of California. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States Government or the University of California, and shall not be used for advertising or product endorsement purposes.

Work performed under the auspices of the U. S. Department of Energy by the University of California Lawrence Livermore National Laboratory under Contract W-7405-Eng-48.

This report has been reproduced
directly from the best available copy.

Available to DOE and DOE contractors from the
Office of Scientific and Technical Information
P.O. Box 62, Oak Ridge, TN 37831
Prices available from (423) 576-8401
<http://apollo.osti.gov/bridge/>

Available to the public from the
National Technical Information Service
U.S. Department of Commerce
5285 Port Royal Rd.,
Springfield, VA 22161
<http://www.ntis.gov/>

OR

Lawrence Livermore National Laboratory
Technical Information Department's Digital Library
<http://www.llnl.gov/tid/Library.html>

Dynamic Response of a Pulse-Heated, Thick-Walled Hollow Sphere: Validation of Code Numerics (U)

Robert E. Canaan
Lawrence Livermore National Laboratory

Volumetric pulse heating of a thick-walled hollow sphere is numerically investigated. The primary objective is to validate a variety of LLNL 3D hydrocodes for modeling the dynamic behavior of fissile/fissionable metals subject to rapid 'fission-heating' transients. The 3D codes tested include both DYNA3D and NIKE3D, as well as the 'ASCI' code, ALE3D. The codes are compared 'head-to-head' and are benchmarked against a 1D finite difference solution to the problem that is derived from basic principles. Three pulse-heating transients are examined with full-width-half-maximum pulse durations of 41 μ s, 85 μ s, and 140 μ s, respectively. These three transients produce a significant range of dynamic responses in the thermo-elastic regime. We present results for dynamic radial displacements and stresses for each pulse, and also discuss which code features/options worked best for these types of calculations. In general, the code results are in excellent agreement for the simple system considered. Validation of code numerics in simple systems is a key first step toward future application of the codes in more complicated geometries (U).

Dynamic Response of a Pulse-Heated, Thick-Walled Hollow Sphere: Validation of Code Numerics (U)

Robert E. Canaan
Lawrence Livermore National Laboratory

1.0 Introduction

When fissile/fissionable metals are exposed to either an internal or external neutron source, fission heating and subsequent thermal expansion of the material can occur. Under certain conditions, the rate of fission energy production may follow a temporal 'pulse' in thermal power. For example, in the supercritical operation of a fast burst nuclear reactor, the power rises sharply following a positive reactivity insertion, but then decreases rapidly due to negative reactivity feedback associated with the expansion of the system. Or, if a fissile/fissionable system is subcritical, pulse heating may result if the system is exposed to a significant burst of external neutrons.

In either case, if the resulting heating pulse occurs rapidly enough, the temperature of the fissionable metal can rise faster than the material can respond by thermal expansion. In other words, there is a lag between the rise in temperature and thermal displacement of the material boundaries. Under such conditions, a portion of the thermal energy is converted to kinetic energy, producing vibrational displacements and potentially large dynamic stresses. This behavior is described in more detail as follows.

Early on in the pulse-heating transient, the mass-inertia effect mentioned above implies that while the material density remains essentially constant, there is an increase in material pressure as the temperature rises. Hence, the fissionable material is initially subject to a compressive stress. Later on, if the material remains elastic, the initial compression gives way to tension as the material elastically rebounds and also begins to finally expand in response to increasing temperature. After the heating pulse is complete and the total temperature rise is achieved, the expanded metal reaches a maximum displacement and peak tensile stress. Here, the dynamic expansion of the fissionable part exceeds the static expansion that would occur if the material were heated slowly. Furthermore, if the stresses remain below the tensile yield, another elastic rebound occurs, again sending the material into compression and contracting the material to a minimum expansion that is below the static value. If the material remains in an elastic regime throughout the transient, this 'ringing' behavior will continue at the natural vibrational frequency of the part in question until it is eliminated by viscous damping. The final displacement of the material

failures and even reactor disassembly have been reported for fast burst reactors subject to pulse heating transients [1, 2]. Secondly, the dynamic displacements that result from such transients can lead to additional stresses if the heated parts are externally constrained. Also, the displacement of the fissionable material surfaces can provide a coupling mechanism between the hydrodynamic and neutronic aspects of the problem.

Early investigations of pulse-heating transients from neutron exposure have been reported by Burgreen [3], who developed analytical expressions describing the dynamic behavior of thin 1D cylindrical and spherical shells. His method involves the replacement of temperature-induced stresses with equivalent body forces, surface tractions, and internal pressures. Austin [4] expanded upon the work by considering the dynamic response of thin, nested spherical shells. Analytical expressions were developed for the case of a heated, elastic inner shell concentric within a plastic, unheated outer shell. The work of both Burgreen and Austin show that the dynamic stress amplitude in a pulse-heated body depends upon the magnitude of the temperature rise as well as the ratio of the heating time to the natural vibrational period for the body. Later, Reuscher [5] developed both 1D and 2D finite difference techniques for analyzing the dynamic pulse behavior of the Sandia Pulsed Reactor II (SPR-II) fast burst reactor, essentially a hollow cylinder comprised of highly enriched uranium alloyed with 10-wt% molybdenum (U-10%Mo). Reuscher's techniques are not limited to thin shells and use a temperature rise function which is a proper representation of the fission-induced power pulse. These added complexities are not easily used with analytical methods. Recently, DiPeso [6] used some of Reuscher's experimental results [5] to validate the 3D explicit hydrocode DYNA3D [7] for the case of a fissile U-10%Mo solid rod irradiated by neutrons in the SPR-II. DiPeso [6, 8] also compared DYNA3D predictions to an analytical result for the case of a thin spherical shell subject to a linear ramp in temperature up to a constant, uniform value. DiPeso included the effects of plasticity in the calculations and found that for extremely short heating pulses, dynamic stresses exceed yield, and that subsequent thermal displacement is greatly diminished by plastic flow.

In this paper, the dynamic behavior of a pulse-heated, thick-walled hollow sphere is examined using four distinctly different numerical techniques. The arbitrarily-sized fissile sphere is computationally subjected to pulse-heating transients with full-width-half-maximum (FWHM) heating times of 41 μ s, 85 μ s, and 140 μ s, which produce a wide range of dynamic responses. The primary objective of the study is to validate a variety of LLNL 3D hydrocodes for modeling dynamic fission-heating types of transients. The 3D codes tested include the engineering codes DYNA3D [7] and NIKE3D [9], as well as the 'ASCI' code, ALE3D [10]. The codes are compared 'head-to-head' and are benchmarked against a 1D finite difference solution to the problem that is derived from Reuscher's early work [1]. The test problem chosen for the present study involves a variety of simplifying assumptions (discussed in the next section) in order to limit the validation to one of code numerics, as opposed to a full validation with direct comparison to experiment. As DiPeso points out, code validation in simple geometries "is a necessary step if calculations involving more complicated geometries are to be understood and trusted [6]." It is also hoped that by simplifying the problem we can gain additional insight into the pulse-fission-heating problem without diluting the fundamental physics.

Another objective of this work is to detail requirements/suggestions for successful application of each of the individual codes to this type of problem. To that goal, this report includes discussions of not only what techniques worked for each code tested, but also what techniques did not work. Sample input decks and limited code listings are included in the Appendices for cases of particular interest.

Section 2.0, which follows, discusses many of the simplifying assumptions used to formulate the test problem. It also provides a general discussion of the four codes used to solve the problem. In Section 3.0, a variety of results are presented as we compare code-to-code predictions for three pulse transients and discuss in detail what models/code features were most successful. Section 4.0 offers concluding remarks and proposals for additional work.

2.0 Methodology

2.1 Problem Overview

Numerical analysis of fission-heated mechanical systems typically begins with some sort of neutronics code that is capable of modeling neutron transport within the fissionable component of interest. If external neutron sources are present, the neutronics model may be additionally required to model neutron transport up to the boundaries of the fissionable part. Either way, the ultimate goal of such an analysis will be to determine the spatial and temporal history of the fission energy deposition in the fissile/fissionable component of interest. With this information in hand, the energy source term can then be directly coupled to the material response equations, i.e. the hydrodynamic problem, such that the resulting material displacements and thermal/mechanical stresses can be determined subject to the appropriate initial and boundary conditions.

However, an important aspect to keep in mind when performing such an analysis is the highly coupled nature of this problem. Note that the spatial and temporal temperature distributions that result from fission heating will be highly dependent upon the neutronic characteristics of the problem. Case in point: if slow (thermal) neutrons externally irradiate a fissile component, a large portion of the fission energy deposition will be near the surface of the component; whereas, if fast neutrons are involved, the heating will be more volumetric. Yet, the nature of the heating itself may have important repercussions on the neutronics problem. For example, the fission-induced temperature changes in the nuclear material can significantly alter the material's neutron absorption capabilities through processes such as Doppler broadening [11], which may promote either more or less fission energy deposition depending upon material composition.

In addition to neutronic coupling through the thermodynamic aspects of the problem, there can also be direct coupling between the neutronics and the hydrodynamics. For example, the dynamic expansion and stress in the fission-heated component is highly dependent upon the spatial and temporal temperature/energy distribution, and thus the system neutronics as described above. Yet the expansion of the fissionable material itself provides neutronic feedback, such as the loss of fission neutrons that often results from an

increase in the part's surface area and/or decrease in material density. A loss of fission neutrons would of course influence the resulting fission energy deposition, and in the case of many fast burst reactors, this mechanism provides the reactor shutdown capability.

Given the complexities described above, it is often desirable to decouple aspects of the pulse fission-heating problem for computer code validation purposes. In the analysis that follows, the neutron transport/kinetics problem will be ignored, and rather, attention will be focussed on the hydrodynamics problem of calculating the dynamic expansions and stresses that result when a hollow sphere is subjected to a variety of pulse-heating transients. In place of a neutronics model, a temperature or energy-rise function will be used which closely approximates the time-dependent energy deposition that occurs in pulse-heated fissile systems such as fast burst reactors. Additionally, heat transfer will be neglected since the dynamic effects of interest here typically disappear before there is time for heat transfer to significantly affect the initial temperature distribution. It will also be assumed that the spatial temperature distribution remains uniform throughout the hollow sphere.

The test problem considered here is a thick-walled hollow sphere of U-10%Mo, heated uniformly by a volumetric source that produces a nonlinear temporal pulse in thermal power. The inner sphere radius is arbitrarily taken to be 5.080 cm (2.0 in) with an outer radius of 7.620 cm (3.0 in). For simplicity, the various material parameters required by the codes are assumed to be both time and temperature independent. Table 1 gives a brief description of the problem including the material properties used for all numerical analyses reported here. The material properties of Table 1 were obtained from Reuscher [12] and are reasonably consistent with literature values for U-10%Mo at room temperature.

TABLE 1. Test Problem Description

parameter	value
inner radius	5.080 cm (2.0 in)
outer radius	7.620 cm (3.0 in)
material	U(93%U ²³⁵)-10%Mo
density	17.2 g/cc
Young's Modulus	9.0e+11 dynes/cm ² (1.31e+04 ksi)
Poisson's Ratio	0.3
Yield stress at offset	9.0e+09 dynes/cm ² (130.5 ksi)
Ultimate stress	9.4e+09 dynes/cm ² (136.3 ksi)
Linear CTE (α)	1.4e-05 °C ⁻¹
Thermal conductivity	0.06 cal/s-cm-°C
Specific Heat (C_p)	0.028 cal/g-°C

We investigate a total of three heating pulses. The individual pulses are characterized by the maximum temperature rise they produce and by the duration of the pulse. The pulse duration is typically expressed as a full-width-half-maximum (FWHM), or pulse width at one-half maximum power. For each of the three pulses considered here, a maximum temperature rise of 420°C is specified, with pulse durations (FWHM) of 41μs, 85μs, and 140μs, respectively. These pulses are characteristic of those seen in fast burst reactor operations [2]. The pulse amplitude (or peak power) is adjusted such that the total energy deposited in the heated material remains the same for all three pulses. Additionally, the time to peak power is staggered. The 41μs-FWHM pulse peaks at 100μs; the 85μs-FWHM pulse peaks at 300μs; and the 140μs-FWHM pulse peaks at 500μs. These pulses produce transient behavior with peak tensile/compressive stresses in the U-10%Mo sphere that are well below the reported yield stress. Thus, only elastic material response is covered in this study.

The transients described above are calculated using four computer codes, each having unique characteristics. Each of these codes will be described in some detail in the next subsection. The first code to be discussed is important in that it will serve as a benchmark for the others. This code is a 1D finite difference approximation to the equations of motion in a spherical geometry. Analytic solutions are difficult to obtain for thick-wall spheres that are heated nonlinearly in time. The remaining codes are all LLNL 3D hydrocodes and include two engineering codes {DYNA3D and NIKE3D} and one 'ASCI' code {ALE3D}.

2.2 Calculational Models

2.2.1 One Dimensional Finite Difference Model

The 1D calculational approach described below is based on the early work of Reuscher [1], who developed the technique for pulse-heated cylinders, spheres, and slabs. However, some errors in Reuscher's derived equations were noticed for the spherical case [1, 5]. Thus, some time will be spent here redeveloping the basic equations in spherical geometry.

For a sphere subject to a time-dependent temperature rise that is a function of the radial coordinate only, the general form of the 1D displacement equation is

$$\frac{\partial^2 u}{\partial r^2} + \frac{2}{r} \frac{\partial u}{\partial r} - \frac{2u}{r^2} - \frac{(1+\nu)}{(1-\nu)} \alpha \frac{\partial T}{\partial r} = \frac{1}{c^2} \frac{\partial^2 u}{\partial t^2}, \quad (\text{EQ 1})$$

where $u(r, t)$ is the radial displacement of the sphere; ν is Poisson's ratio; α is the linear coefficient of thermal expansion (CTE); $T(r, t)$ is temperature; c is the speed of sound; and r and t are the radial and time coordinates, respectively. The above equation assumes elastic material behavior, with time and temperature independent properties.

Equation 1 is solved for the radial displacements, $u(r, t)$, subject to a specified spatial and temporal variation of temperature, $T(r, t)$. By specifying the temperature function in this manner, the hydrodynamic problem is decoupled from the neutronic problem. After the

displacements are determined via Equation 1, the stress components throughout the sphere are calculated using the following relations for the radial and tangential (or hoop) stresses, respectively [13]:

$$\sigma_{rr}(r, t) = (2\mu + \lambda) \frac{\partial u}{\partial r} + \frac{2\lambda u}{r} - (3\lambda + 2\mu) \alpha T(r, t) \quad (\text{EQ 2})$$

$$\sigma_{\theta\theta}(r, t) = \lambda \frac{\partial u}{\partial r} + \frac{(2\mu + \lambda)}{(1 - \nu)} \frac{u}{r} - (3\lambda + 2\mu) \alpha T(r, t) , \quad (\text{EQ 3})$$

where

$$\lambda = \frac{\nu E}{(1 + \nu)(1 - 2\nu)} , \quad (\text{EQ 4})$$

and

$$\mu = \frac{E}{2(1 + \nu)} , \quad (\text{EQ 5})$$

and E is Young's Modulus.

Equation 1 must be solved subject to an initial condition. Here, we specify that the sphere is at rest at time zero, in other words the displacement components throughout the body are all zero at time zero:

$$u(r, 0) = 0 \quad \text{at } t = 0. \quad (\text{EQ 6})$$

The boundary conditions for Equation 1, since our sphere is hollow, specify that the inner and outer surfaces are free boundaries. This is akin to requiring that the radial or normal stress at each surface is zero, or

$$\sigma_{rr}(B, t) = 0 , \quad (\text{EQ 7})$$

where B is the radial coordinate of the surface. By applying Equation 7 with Equation 2, the boundary condition equation(s) can be written completely in terms of the boundary displacements, i.e.

$$(2\mu + \lambda) \frac{\partial u(r = B, t)}{\partial r} + \frac{2\lambda u(r = B, t)}{B} = (3\lambda + 2\mu) \alpha T(r = B, t) . \quad (\text{EQ 8})$$

Reuscher [1] showed that the transient behavior of the system is sensitive to the function used to describe the temperature rise. He considered in Ref. [1] several temporal-temperature functions that are commonly used to describe the pulse behavior of fission-heated systems, including a linear ramp to a constant temperature, a cosine function, and a 'burst-integral' function. Significant differences in the predicted maximum displacements and stress were noted between the various functions. The function that closest approximates the temperature rise during the power pulse of a fast burst reactor is the

'burst-integral' function, which is proportional to the integral of a symmetric power pulse, and is given by:

$$T(t) = \frac{T_{max}}{\exp\left[-\frac{3.52}{b}(t-t_{pp})\right] + 1}, \quad (\text{EQ 9})$$

where T_{max} is the maximum temperature rise; b is the full-width-half-maximum (FWHM); and t_{pp} is the time to peak power [14]. Since the heated sphere is effectively adiabatic over the time period of interest here, the specific power may be found by differentiating Equation 9, i.e.

$$q(t) = C \frac{dT}{dt} = \frac{\frac{3.52CT_{max}}{b} \exp\left[-\frac{3.52}{b}(t-t_{pp})\right]}{\left\{ \exp\left[-\frac{3.52}{b}(t-t_{pp})\right] + 1 \right\}^2}, \quad (\text{EQ 10})$$

where q is the specific power and C is the specific heat (energy/mass-temperature). Figure 1 plots Equations 9 and 10 for the first pulse transient that will be considered in this study, a $41\mu\text{s}$ FWHM pulse that peaks at $100\mu\text{s}$ and results in a maximum temperature rise of 420°C .

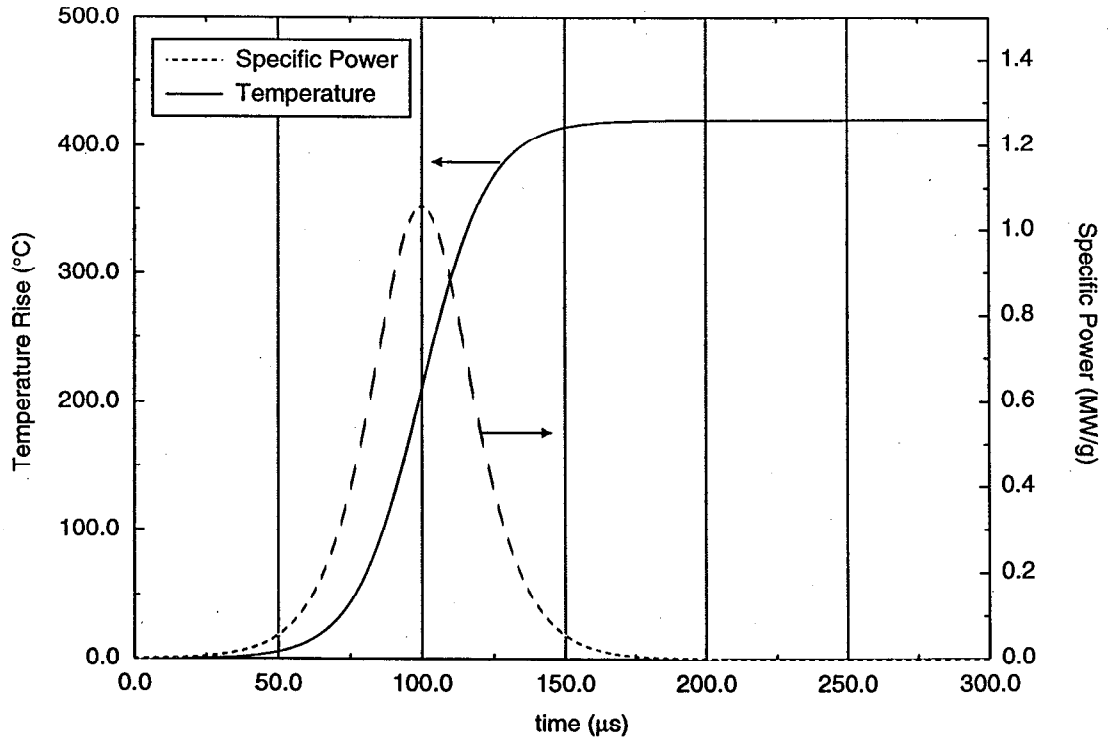


Figure 1. Time history of temperature and specific power for the $41\mu\text{s}$ FWHM pulse, with $\Delta T = 420^\circ\text{C}$.

The method used to numerically solve Equation 1 is an explicit finite-difference technique in which the spherical shell is subdivided into equally-spaced radial zones with nodes located directly on the boundaries. For the interior nodes, a second-order central-difference scheme is used to approximate the partial derivatives. The temperature is taken to be spatially uniform with a time variation specified by Equation 9. After the interior displacements are calculated for a given time step, the boundary node displacements are calculated using Equation 8, written in either forward or backward second-order differences depending upon whether the boundary is an inner or outer surface. After the displacements have been calculated across the body for a given time step, the stress components are determined from the second-order finite-difference forms of Equations 2 and 3, again using forward/backward forms for the boundary nodes and central differencing for the interior nodes. According to Reuscher, the method is stable providing the time step, Δt , satisfies the following Von Neumann criteria:

$$\Delta t^2 \leq \frac{2}{\frac{2c^2}{\Delta x^2} + \frac{2c^2}{r_{i,min}^2}}, \quad (\text{EQ 11})$$

where Δx is the spatial increment; c is the speed of sound; and $r_{i,min}$ is the minimum value of the radial coordinate for a hollow sphere. Equation 11 implies that the time step must be smaller than the time required for a stress wave to propagate across a single spatial increment.

The 1D finite-difference solution to the above relations is included in Appendix A as a code listing. The code is given the name 'newpulse.f' and is programmed in FORTRAN. This code may be applied to predict the dynamic displacements and stresses for either a pulse-heated hollow sphere or hollow infinite cylinder. Also, the code is not limited to a spatially uniform temperature, and the user may program a spatial temperature function that is appropriate for the system of interest. The numerics of the 'newpulse.f' code have been validated against the numerical data presented by Reuscher in Refs. 1 & 5 for the case of a pulse-heated hollow *cylinder*. Specifically, 'newpulse.f' was used to reproduce the 1D finite-difference solution that Reuscher presents for a 41 μ s FWHM, 420°C pulse in the SPR-II geometry. Because analytic solutions [3, 4] to this problem are limited to thin shells, we will use the 1D finite-difference solution discussed above as a 'benchmark' solution to assess the performance of a variety of more sophisticated 3D codes. The overall characteristics of the various 3D codes are discussed in the subsections that follow.

2.2.2 DYNA3D

DYNA3D [7] is an explicit, Lagrangian hydro/structural mechanics code for analyzing the transient dynamic behavior of three-dimensional solids and structures. The code uses a finite-element formulation to discretize 3D space and a finite-difference time discretization scheme. DYNA solves the complete set of material response equations and the energy equation without the approximations that lead to Equation 1. DYNA additionally has a wide range of available material models. In this analysis, Material Model #4 is used, which is a thermo-elastic-plastic constitutive model associated with a

one-term equation of state (EOS) that relates the pressure rate to the total strain rate via a bulk modulus. The total strain is determined by adding/subtracting the thermal strain (i.e. $\alpha\Delta T$) to the calculated mechanical strain. Although material properties for this material model can be fully temperature dependent, we again stick with the temperature-independent parameters given previously in Table 1. Also, the effects of plasticity are ignored.

DYNA Material Model #4 requires the specification of a coefficient of thermal expansion (CTE or α). The CTE must be defined with respect to a reference temperature, T_{ref} , for a given material at the start of the calculation. In the calculations that follow, the reference temperature is taken to be 0°C, which is also the uniform material temperature at time zero. It is important to note that in DYNA, the specified CTE must be a linear secant value as opposed to a linear tangent value. Many literature sources report the tangent CTE as a function of temperature. The two CTE representations are related as follows:

$$\sim\alpha_{DYNA} = \frac{1}{T - T_{ref}} \int_{T_{ref}}^T \alpha(T) dT \quad (EQ 12)$$

where

α_{DYNA} is the secant or average CTE required by DYNA,

and

$\alpha(T)$ is the tangent CTE at a given temperature.

Note that for the present case of temperature-independent properties, the two CTE values given above are equivalent, i.e. $\alpha_{DYNA} = \alpha$.

Since Material Model #4 is temperature dependent, the thermal effects option in Control Card #6 must be non-zero. This card allows the temperature-time history to be specified either via a load curve (a table), or by reading nodal temperatures from an external data file. In this analysis, the former option is used in which the nodal temperature for all nodes in the mesh is taken to be spatially uniform. This load curve is included in the DYNA input deck, and corresponds to a table of time vs. temperature values. These tables are generated in accordance to Equation 9 for each of the pulse transients considered. A sufficient number of entries are included in the table to accurately represent the desired thermal transient.

A one-eighth-symmetry 3D mesh of a hollow sphere is created using the TRUEGRID [15] mesh generator, using hexahedral eight-node elements. TRUEGRID meshes are used for all the 3D codes evaluated here. The initial conditions for this problem are specified by initializing all nodal velocities to zero. Symmetry boundary conditions on the x, y, and z faces of the hollow sphere are applied by constraining nodal motion normal to the respective symmetry planes. This is akin to allowing the inner and outer surfaces of the sphere to have free radial motion. Structural and/or numerical damping with time is not allowed. As DYNA is an explicit code, the stable time step is governed by the Courant

limit. DYNA automatically determines this maximum time step at each step in the solution process, and accordingly adjusts the time step to minimize the total number of steps which must be taken. A sample DYNA3D input deck for this problem (excluding the node and zone definition sections) is attached as Appendix B. A version of the DYNA3D executable is used which allows the time-dependent plotting of nodal temperature in lieu of x-acceleration. All DYNA3D results presented here were obtained with Version 4.0.14.

2.2.3 NIKE3D

NIKE3D [9] is an implicit Lagrangian structural-mechanics code also based on the finite-element method for 3D spatial discretization. Because the code uses implicit time integration, it is considerably more efficient for static or low rate dynamic problems. Thus, it is not well suited to the class of problems considered here, which deal with shock-induced dynamics over microsecond time scales. However, the code was useful in this validation study in that it offered a numerical approach to the problem that was unique to the four codes tested (the other codes are all explicit). Interestingly enough, NIKE3D proved capable of providing accurate predictions for all pulse-heating transients considered, although there is certainly no advantage to using an implicit scheme here since the time step must be small enough to resolve the dynamic frequencies of interest. Because the implicit time integrator is unconditionally stable, the user must determine the appropriate time step for the problem at hand. Here, we use a fixed time step of $1.0\mu\text{s}$ for all three pulse transients considered with good accuracy. Also, the computational burden associated with implicit schemes is considerably greater than with an explicit approach. However, the NIKE3D computation times in this study were reduced by running the problems with a linear solver and using multiple processors.

As was the case with the DYNA3D analysis, a thermo-elastic material model is required for the NIKE3D simulations (also Material Model #4). As with DYNA, the thermal effects option (Control Card #7) is used such that the temperature for all nodes in the mesh is spatially uniform and time varying according to a specified load curve (Equation 9). NIKE3D also expects a secant linear CTE as defined by Equation 12, and additionally requires the specification of a material reference temperature in Card #1 of the Material Deck (set here to 0°C).

All NIKE simulations are run with a new 'incompatible modes' brick-element formulation. The old 'incompatible modes' option results in significant errors for isolated nodes in the problem. At the time of this writing, the new 'incompatible modes' option is invoked as follows. On Control Card #8, the 'brick element formulation' should be set to '11' [as opposed to '12' for the old 'incompatible modes' option]. Additionally, two changes are required for Material Card #1. First, the 'element class' card should be set to '20', and the last four columns of Material Card #1 should contain '1139' [columns 77-80]. A sample NIKE3D input deck is attached as Appendix C for reviewing these cards.

To reduce the computation times associated with running NIKE3D with a $1\mu\text{s}$ timestep, these problems are run with a linear solver on three processors. The former is accomplished by suppressing both equilibrium iterations (which are required for nonlinear problems) and stiffness matrix reformations. See Control Card #6 in Appendix C, where

the time steps between equilibrium iterations and stiffness reformations are set greater than or equal to the total number of time steps in the problem. To run the problems on three processors, the 'PSUB' batch-submission script file should contain the line:

```
setenv MP_THREAD_COUNT 3
```

Other useful NIKE3D options include setting the 'acceleration data dump flag' on Control Card #5 to a value of '3'. This allows displacements to be directly calculated by NIKE3D, as opposed to having the displacements calculated within a post processor. This flag also allows the time-dependent plotting of nodal temperature, which is stored under 'x velocity'. Also, to suppress numerical and structural damping, the default values for the Newmark integration parameters (Control Card #7) and the Rayleigh damping parameters (Material Card #1) are used. All NIKE3D results presented here are from Version 3.3.5.

2.2.4 ALE3D

ALE3D [10] is a 3D arbitrary-Lagrangian-Eulerian hydrocode system being developed at LLNL as part of the Accelerated Strategic Computing Initiative (ASCI). ALE3D has both explicit and implicit hydrodynamics capabilities, as well as a variety of ancillary models that can be used to couple thermal diffusion, thermal radiation, and reaction chemistry to the continuum mechanics problem. ALE3D is finite-element based, and may be run in either a purely Lagrangian mode, a purely 'Eulerian' mode, or in an ALE mode. For the latter, the basic computational step consists of an initial Lagrangian step followed by a remap step in which the computational mesh is 'relaxed' to avoid mesh entanglement or distortion. During the mesh remapping step, advection of mass, momentum, and energy occurs across element boundaries. If the element boundary coincides with a material boundary, the advection step produces a mixed or multi-material element.

In this paper, ALE3D is used in a purely-Lagrangian fashion since material displacements are expected to be small with minimal mesh distortion. The explicit hydrodynamics model is used both alone and in conjunction with the thermal diffusion package to simulate the dynamic behavior of the pulse-heated sphere. We also make use of ALE3D's Engineering Material Model #54, which utilizes a modified Gruneisen EOS in conjunction with a thermo-elastic-plastic constitutive model. Unlike the one-term EOS used in conjunction with Material Model #4 (DYNA & NIKE), the Gruneisen EOS allows full coupling and hydrodynamic feedback between material pressure and energy/temperature. ALE3D's Material Model #54 requires the tabular input of initial yield stress; Young's modulus; Poisson ratio; CTE; plastic modulus; and specific heat as functions of temperature. Again, to be consistent with our previous code simulations, temperature-independent properties are used (see Table 1) and the effects of plasticity are ignored.

With regard to material property input for this material model, a couple of important points need to be made. First, ALE3D expects a volumetric as opposed to a linear coefficient of thermal expansion (CTE). In general, the two are related as follows:

$$\beta(T) = 3\alpha(T) \quad (\text{EQ 13})$$

where

β is the volumetric CTE required by ALE3D,

and

α is the linear tangent CTE.

Equations 12 and 13 can be combined in order to relate the CTE required by ALE3D to that required by DYNA3D:

$$\beta(T) = 3 \left\{ \alpha_{DYNA} + (T - T_{ref}) \frac{d}{dT} \alpha_{DYNA} \right\}. \quad (\text{EQ 14})$$

Secondly, ALE3D uses a specific heat at constant volume, C_v , to relate zonal temperature and energy. However, during material property input for Model #54, the keyword 'specificheat' refers to a specific heat at constant pressure, C_p . The two specific heats are related in the code as follows:

$$C_v = \frac{C_p}{1 + \gamma \beta T}, \quad (\text{EQ 15})$$

where γ is a Gruneisen EOS parameter, β is the volumetric CTE, and T is temperature.

In the present approach, a temperature-*independent* specific heat is desired in order to achieve consistent results with earlier models. However, initial specification of that parameter using the 'specificheat' keyword resulted in a temperature-*dependent* specific heat to be defined according to Equation 15. Thus, a relationship between zonal energy and temperature was established that was inconsistent with previous approaches. Once this problem was realized, it was circumvented by *directly* specifying the desired specific heat as C_v using the 'tablcv' keyword. ALE3D users utilizing Engineering Material Model #54 should keep the relationship of Equation 15 in mind when providing specific heat data to the code.

ALE3D is an energy rather than a temperature-based code. Its hydrodynamic equations of state additionally allow coupling between material energy/temperature and dynamic changes in volume/pressure. Thus, for the majority of the transient cases considered here, an energy-deposition (rate) load curve rather than a temperature load curve is used to define the ALE3D fission pulse, i.e. Equation 10 is used rather than Equation 9. This is accomplished in ALE3D using the 'heatgen fixed' keyword in the Boundary Block in conjunction with a table of energy rate vs. time values. Note that Equation 10 defines the pulse in terms of a *specific* energy deposition rate (power per unit mass). The power should be normalized by mass, not initial volume, since the volume fluctuates during the course of the transient. The other initial and boundary conditions are specified in a similar manner to that of the other 3D codes. As before, the heating is spatially uniform.

Despite that fact that ALE3D is an energy-based code, the fission pulse can also be specified in the same way it was prescribed with DYNA3D and NIKE3D, i.e. in terms of nodal temperatures vs. time. Here, we use both approaches (i.e. specified energy and specified nodal temperature) for the first transient to be considered. In the second approach, Equation 9 is used to specify the pulse in terms of nodal temperature, as was done with DYNA and NIKE. This is accomplished in ALE3D using the ‘heat temperature’ keyword in conjunction with a temperature vs. time table. Again, there are no spatial temperature gradients. However, unlike the energy-based approach, which requires only a Hydro Block, the nodal temperature approach requires an ALE3D Thermal Block (thermal diffusion model) in addition to the Hydro Block. This is because the ‘heat temperature’ keyword is typically used to prescribe nodal-temperature boundary conditions for the thermal diffusion model. However, in this case it is used to prescribe the temperature history of all of the nodes in the mesh. Furthermore, for this case there must be full coupling between the hydro and thermal solvers. The results for both ALE3D approaches to this problem are presented and discussed in Section 3.0. Sample ALE3D-generator input files for both approaches are included in Appendix D.

All ALE3D simulations are run using Version 2.8.4. Previous versions were discovered to have a ‘bug’ associated with the time-dependent output of nodal displacements when the code was run in a purely Lagrangian manner.

3.0 Results

This section is organized as follows: First, results will be presented that correspond to the one-dimensional finite difference ‘benchmark’ to the problem, described previously in Section 2.2.1. These results will be presented and discussed for the first pulse transient considered, a $41\mu\text{s}$ FWHM pulse with a maximum temperature rise of 420°C . Secondly, the 3D codes tested will be compared ‘head-to-head’ with the finite difference benchmark. Comparisons will be made for dynamic radial displacements and tangential or ‘hoop’ stresses at the inner and outer surfaces of the sphere. The effects of different numerical approaches taken with some of the codes will be examined, as will the issue of mesh independence. A total of three pulse transients are studied, starting with the $41\mu\text{s}$ FWHM case, and then examining longer pulses of $85\mu\text{s}$ and $140\mu\text{s}$.

3.1 1D Finite Difference Results

One-dimensional, finite-difference results are presented for the $41\mu\text{s}$ FWHM case to illustrate some of the physics details associated with pulsed volumetric heating in spherical structures. In this case, the hollow U-10%Mo sphere is subjected to the thermal pulse seen earlier in Figure 1. The $41\mu\text{s}$ FWHM pulse reaches a peak specific power of $1.06\text{MegaWatts/gram}$ at time = $100\mu\text{s}$, producing a maximum temperature rise of 420°C by $200\mu\text{s}$. Since heat transfer is neglected over these time scales, the maximum temperature remains at 420°C for the duration of the calculated transient ($300\mu\text{s}$).

The calculated results for this transient are seen in Figures 2-5. All 1D finite difference results correspond to a nominal timestep of $0.01\mu\text{s}$ and a spatial discretization of 20 nodes

throughout the spherical shell. Convergence of these results was examined by independently varying the timestep (0.1 μ s, 0.01 μ s, 0.001 μ s) and spatial discretization (20, 50, 100 nodes), within the stability limits of Equation 11, to assure that the data are temporally and spatially converged. Also, the nominal second-order difference approximations were replaced with third-order differencing. In all cases, changes in the reported results were not perceptible. Thus, it is concluded that the results reported herein are spatially converged with N = 20 nodes (19 zones).

Figure 2 shows the predicted radial displacements as a function of time for both the inner and outer surfaces of the sphere. By comparing the temperature rise function of Figure 1 to the nodal displacements of Figure 2, one can indeed see the initial inertial lag between the expansion of the sphere and the rising temperature. This is particularly evident of the inner surface, whose internal constraints differ from those imposed on the outer surface. Specifically, the inner surface must expand outward against the inertia of the rest of the spherical shell. Because of this inertial lag, the thermal energy does not go immediately into material expansion, but rather into an initial increase in the material pressure, as seen in Figure 3. This increase in pressure corresponds to a compressive hoop stress at early times, seen in Figure 4. Because of the greater internal constraints on the inner surface, the hoop stresses are greatest at that location.

At roughly 100 μ s, which corresponds to the peak of the power pulse, the elastic material experiences a rebound. Also, the material finally begins to expand as a result of the still increasing temperature. This throws the sphere into a tensile state, with corresponding negative pressures. The peak tensile stress (and maximum negative pressure) occurs at roughly 150 μ s, at which time the peak radial displacement also occurs. As shown in Figure 2, the maximum displacement for both surfaces exceeds the static limit, or the expansion that would result if the material were heated slowly. For the outer surface, the maximum displacement exceeds the static value by ~34%; whereas, for the inner surface the maximum displacement is over 74% greater than the static value. Also, the results show clearly that the critical stress region occurs at the inner surface where the maximum dynamic hoop stress occurs. Here, a peak tensile stress of ~80ksi (80,000psi) is reached, which is still below the elastic yield of ~130ksi. Beyond 150 μ s, another elastic rebound occurs, again throwing the sphere into a compression with a maximum compressive stress (at 200 μ s) of equal magnitude to the maximum tensile stress. The corresponding radial displacements reach a value that is contracted with respect to the static limits. These dynamic, elastic oscillations continue at the natural vibrational frequency of the hollow sphere, given by:

$$\omega = \sqrt{\frac{2(E)}{(1 - \nu) \rho R^2}}, \quad (\text{EQ 16})$$

where ω is the vibrational frequency; E is Young's Modulus, ν is Poisson's Ratio, ρ is density, and R represents the mean radius of the spherical shell. The vibrational period that corresponds to Equation 16 is just,

$$\tau = \frac{2\pi}{\omega}. \quad (\text{EQ 17})$$

For the present geometry, Equation 17 gives a natural vibrational period of $103\mu\text{s}$, which is consistent with the behavior seen in Figures 2-4. Since structural or numerical damping is not included in the calculations, the amplitude of the oscillations does not diminish with time.

Figure 5 plots hoop stress against the total radial strain for this pulse. The behavior of Figure 5 is consistent with the previous figures. Namely, the sphere is initially thrown into a compressive state, with the inner surface hoop stress exceeding the compressive stress of the outer surface by roughly a factor of 2. By nature of the geometry, the inner surface has a smaller strain at this time compared to the outer surface. Then, as the compression lessens, both inner and outer surfaces pass into tension at the same time and at the same total strain. At the peak tensile state, the inner surface stress again exceeds that of the outer surface by roughly a factor of 2 (80 vs. 40ksi, respectively). As the oscillations continue in the elastic regime, both surfaces move along the same stress/strain line of equal slope, corresponding to the specified elastic modulus.

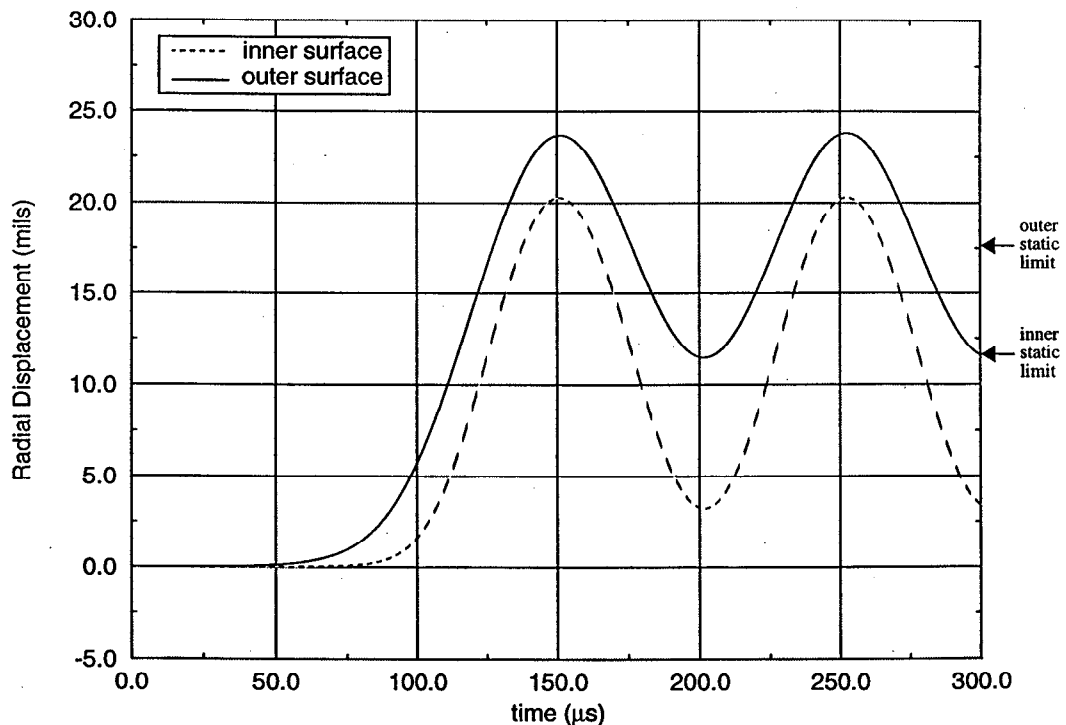


Figure 2. Radial displacement history of inner and outer surfaces for the $41\mu\text{s}$ FWHM pulse, $\Delta T = 420^\circ\text{C}$.
[1D finite difference results with $\Delta t = 0.01\mu\text{s}$, $N = 20$ nodes]

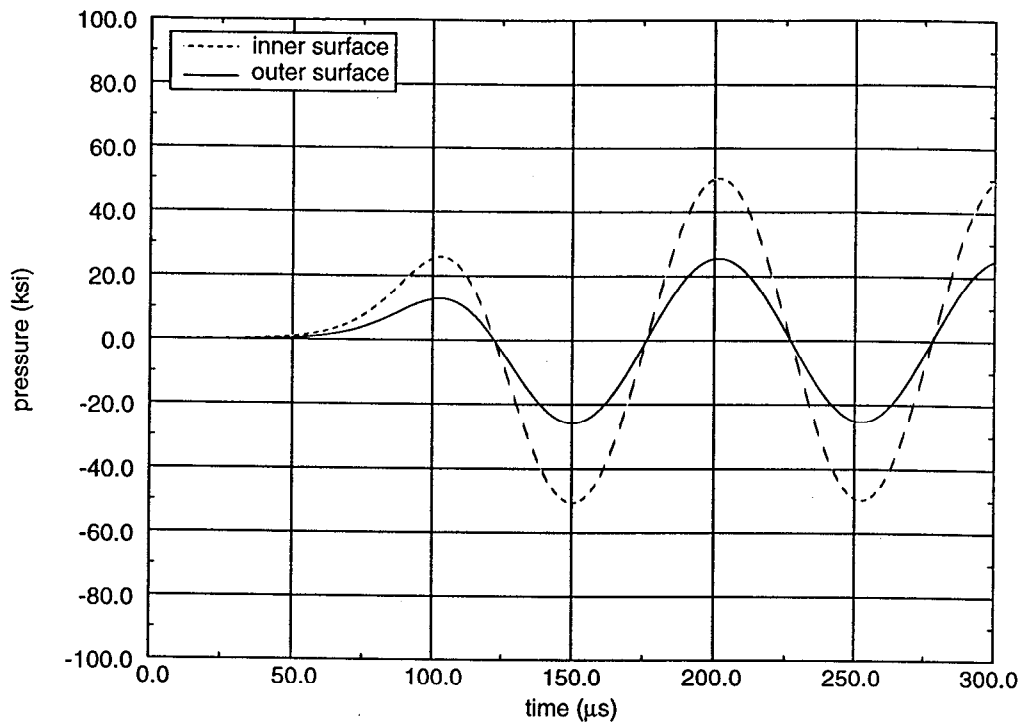


Figure 3. Pressure history of inner and outer surfaces for the 41 μ s FWHM pulse, $\Delta T = 420^{\circ}\text{C}$.

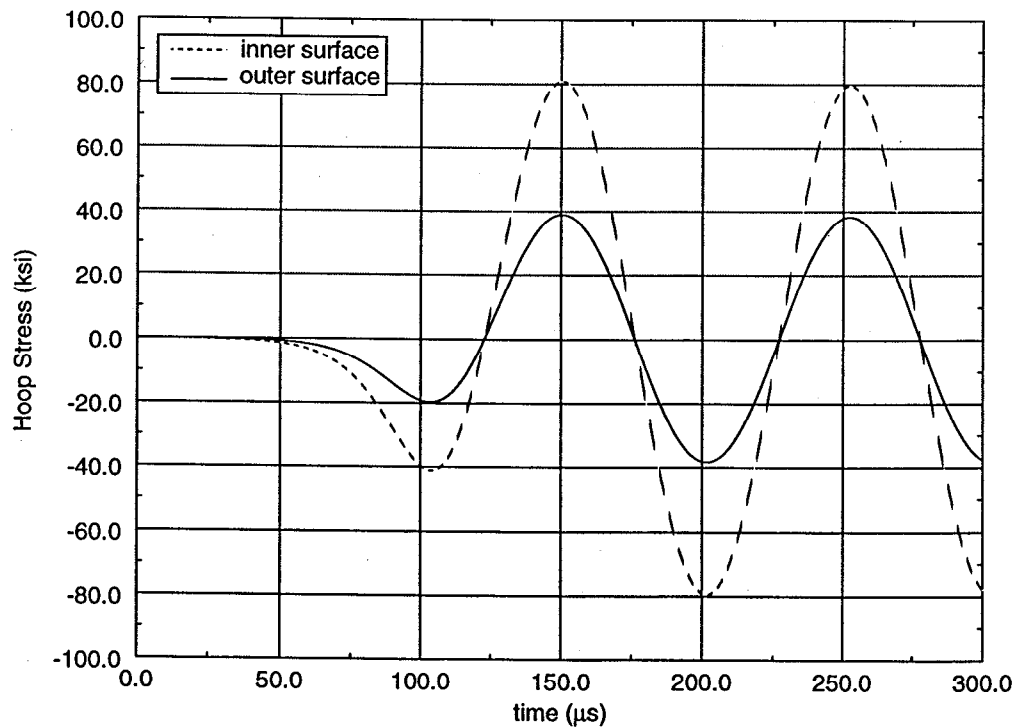


Figure 4. Hoop stress history of inner/outer surfaces for the 41 μ s FWHM pulse, $\Delta T = 420^{\circ}\text{C}$.

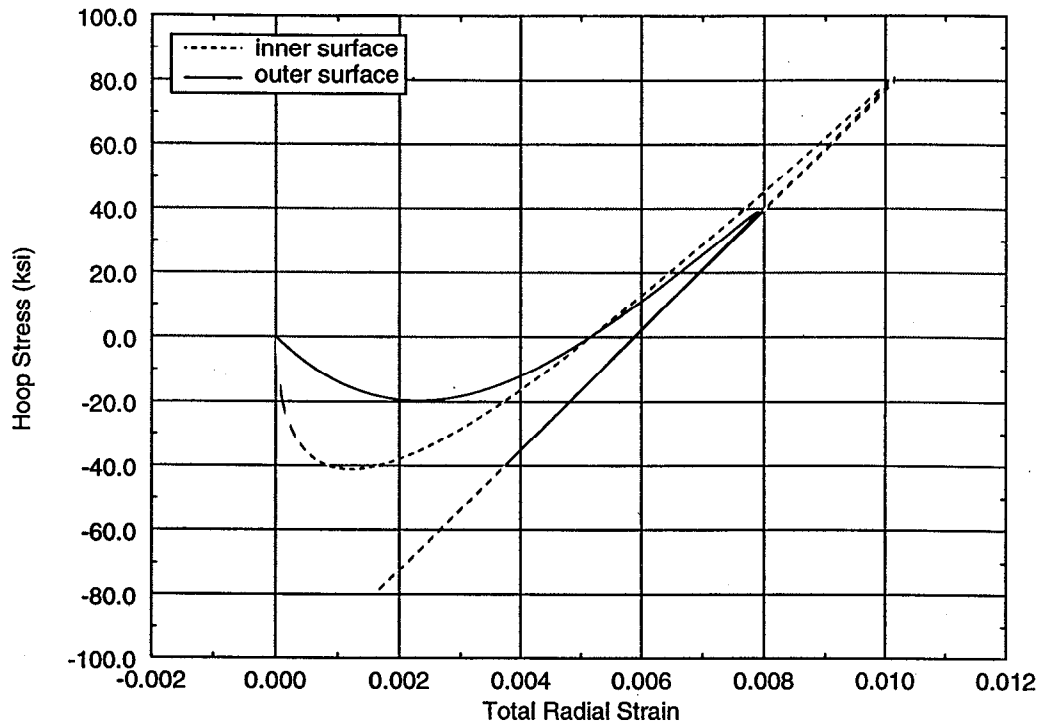


Figure 5. Hoop stress vs. strain for 41 μ s FWHM pulse, inner and outer surfaces.

3.2 Comparison of 3D Codes to 1D Finite Difference Results

3.2.1 41 μ s FWHM, 420°C Pulse

The majority of the 3D pulse-heating simulations were performed on the TRUEGRID-generated mesh shown in Figure 6. This one-eighth-symmetry sphere has an inner radius of 5.08cm (2.0") and an outer radius of 7.62cm (3.0"). This mesh has a total of 20,480 zones with 16 zones through the radial thickness. Symmetry planes exist at $x=y=z=0$, allowing the inner and outer surfaces to behave as free boundaries.

Figures 7 and 8 respectively show the predicted radial displacements and hoop stresses for the first pulse transient. Results from the 3D codes are shown as symbols and the 1D finite difference results are shown as a solid line. Results are presented for both the inner and outer surfaces of the sphere. Additionally, Table 2 compares the predicted maxima/minima values for displacement and hoop stress for both surfaces.

In general, the agreement between the four codes is excellent, especially considering the diversity of the numerical approaches. The overall dynamic trends, vibrational period, and pulse amplitudes are all well predicted. There are some slight discrepancies however. For example, while the radial displacements seem to be very well predicted (cf. Figure 7), there is a small but consistent overshoot in the maximum hoop stress predicted by the 1D finite difference code compared to the 3D finite element codes (cf. Figure 8). This overshoot is primarily noticeable for the inner surface.

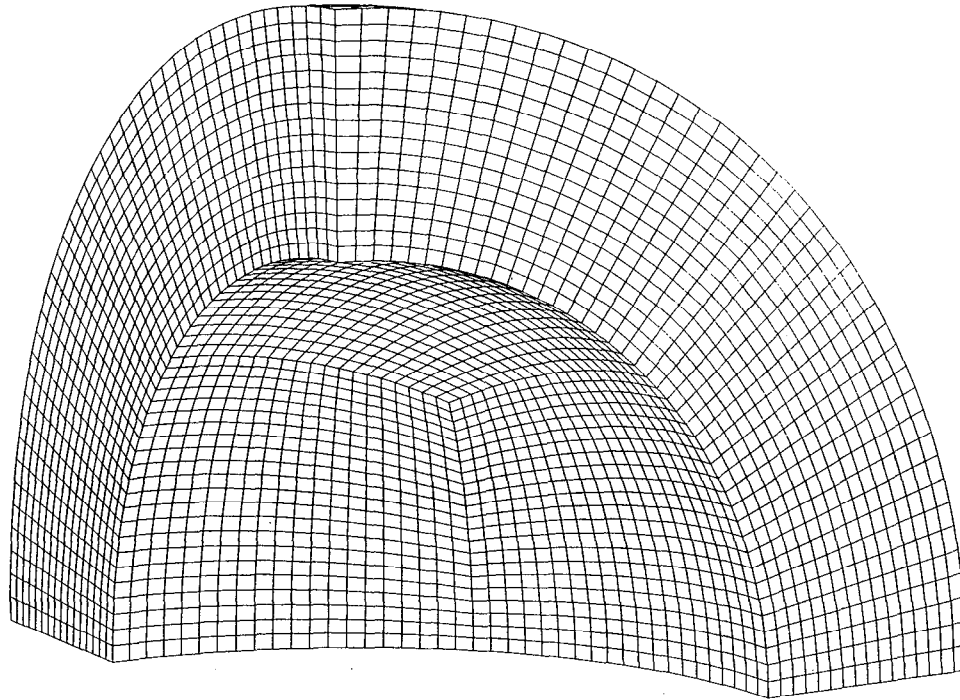


Figure 6. Nominal resolution finite element mesh with 16 radial elements.
[inner radius = 2.0 in., outer radius = 3.0 in.]

TABLE 2. Code Comparison of Max./Min. Displacement and Stress, 41 μ s FWHM, 420°C Pulse

Code	Outer Surface				Inner Surface			
	u_{\max}	u_{\min}	$\sigma_{\theta\theta,\max}$	$\sigma_{\theta\theta,\min}$	u_{\max}	u_{\min}	$\sigma_{\theta\theta,\max}$	$\sigma_{\theta\theta,\min}$
1D, F.D.*	23.7mils	11.5mils	39.0ksi	-38.3ksi	20.3mils	3.3mils	80.9ksi	-79.5ksi
DYNA3D	23.6	11.7	38.6	-37.7	20.1	3.6	76.0	-74.7
NIKE3D	23.5	11.7	38.7	-37.8	20.0	3.5	76.7	-75.1
ALE3D	23.8	11.8	38.7	-38.0	20.2	3.6	75.7	-75.2

* One-dimensional finite difference

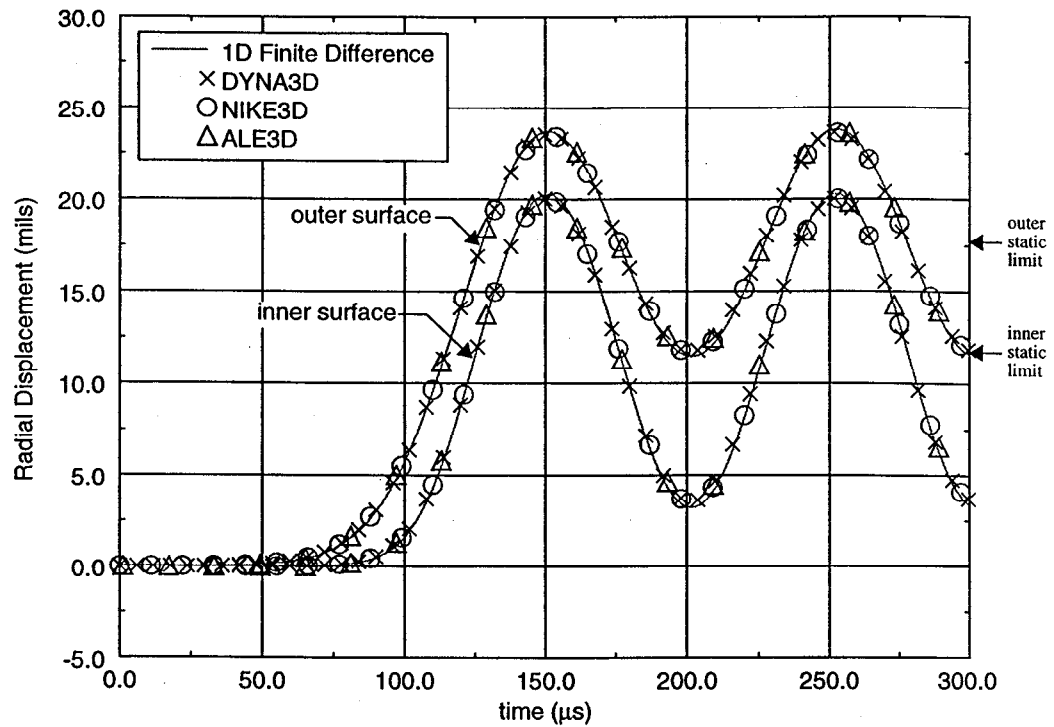


Figure 7. Code comparison of radial displacements for 41 μ s FWHM pulse with $\Delta T = 420^{\circ}\text{C}$, inner and outer surface behavior shown.

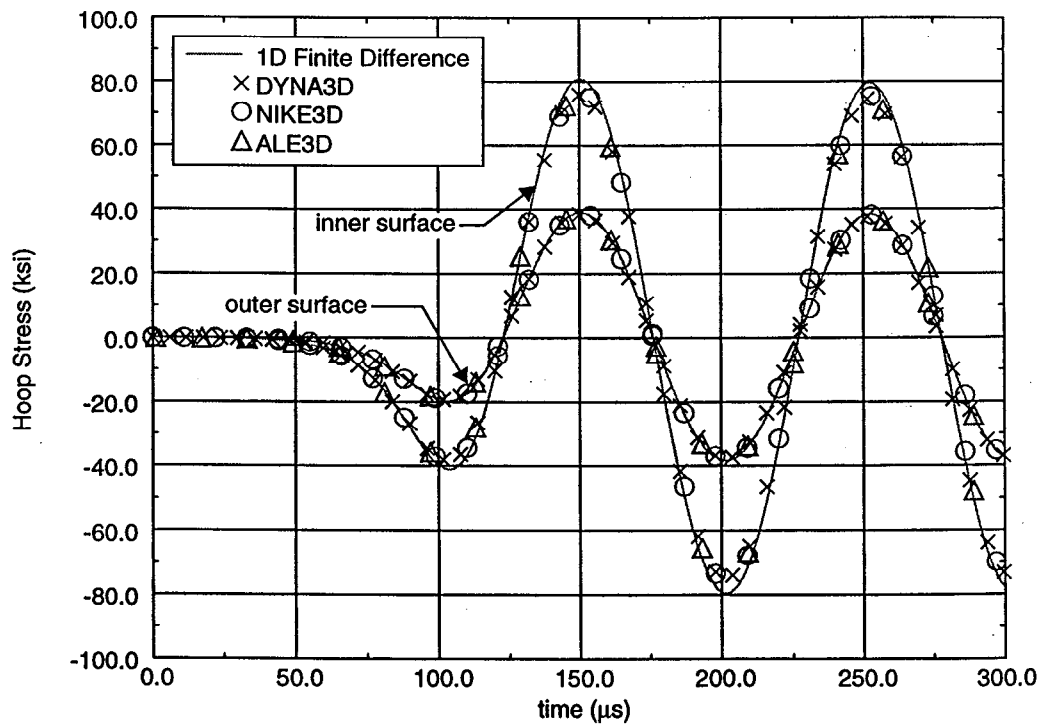


Figure 8. Code comparison of hoop stresses for 41 μ s FWHM pulse with $\Delta T = 420^{\circ}\text{C}$, inner and outer surface behavior shown.

One of the possible reasons for the stress overshoot of the finite difference code involves the distinction between 'zone centered' and 'nodal' variables. For all the finite element codes used in this study, stress is considered a 'zone-centered' quantity. Hence, for DYNA3D, NIKE3D, and ALE3D, the reported hoop stress for the inner and outer surfaces is actually associated with a point that lies just short of the material boundary. However, for the finite difference code, stress is a nodal quantity. Since the 'first' and 'last' radial nodes lie directly on the inner and outer boundaries respectively, the finite-difference-calculated stress at these locations is truly associated with a point *on* the material boundary. On the other hand, radial displacement is a nodal variable for all four codes considered. Hence, it is possible for the four codes to have excellent agreement with respect to predicted displacements, and yet have small discrepancies between the finite element and finite difference methods with respect to predicted stresses. Even more specifically, compared to the finite element codes, the finite difference code should tend to predict a slightly larger stress associated with the inner surface, and a slightly lower stress for the outer surface. This behavior is clearly seen for the case of the inner surface, but for some reason is not evident for the outer surface.

On another note, it should be pointed out that for the ALE3D results shown above, the fission pulse was specified in terms of specific power vs. time (Equation 10), as opposed to temperature vs. time (Equation 9) like the other codes. Nevertheless, it is also possible to specify the ALE3D pulse with a time vs. temperature table by defining all of the nodes in the mesh to be a single nodeset, and then using the 'heat temperature' keyword to assign a temperature table to the nodeset (see Appendix D). The latter approach however requires full coupling between the hydrodynamics solver and the thermal diffusion solver as explained in Section 2.2.4.

Figures 9 and 10 respectively compare ALE3D-calculated results for zonal temperature and displacement, in each case using both an energy table approach ('hydro only') and a temperature table ('hydro + thermal') method. In Figure 9, clearly something is amiss in that the two approaches give different results for zonal temperature. Part of the problem is that ALE3D carries two temperature fields, a zonal temperature (plotted in Fig. 9), and also a nodal temperature. The two temperature fields are used to link the thermal diffusion package to the hydrodynamics model. For the 'hydro only' case, the zonal energy deposition is specified and zonal temperature is calculated directly, and correctly, from energy. However, for the 'hydro + thermal' case, nodal temperature is specified and clearly some error is introduced in converting the nodal temperature field to energy and then back to zonal temperature.

In Figure 9, this results in a 5-10°C overshoot in zonal temperature for the 'hydro + thermal' case compared to the 'hydro only' case (which correctly peaks at 420°C). Figure 9 also shows that both zonal temperatures oscillate slightly due to the adiabatic compression/expansion of the sphere; however, this effect is magnified in the 'hydro + thermal' case. Also, the differences in the predicted zonal temperature between the two approaches lead to small errors in the predicted displacement, as seen in Figure 10. Here, the 'hydro + thermal' case overshoots the 'hydro only' results by roughly 0.5mil and is

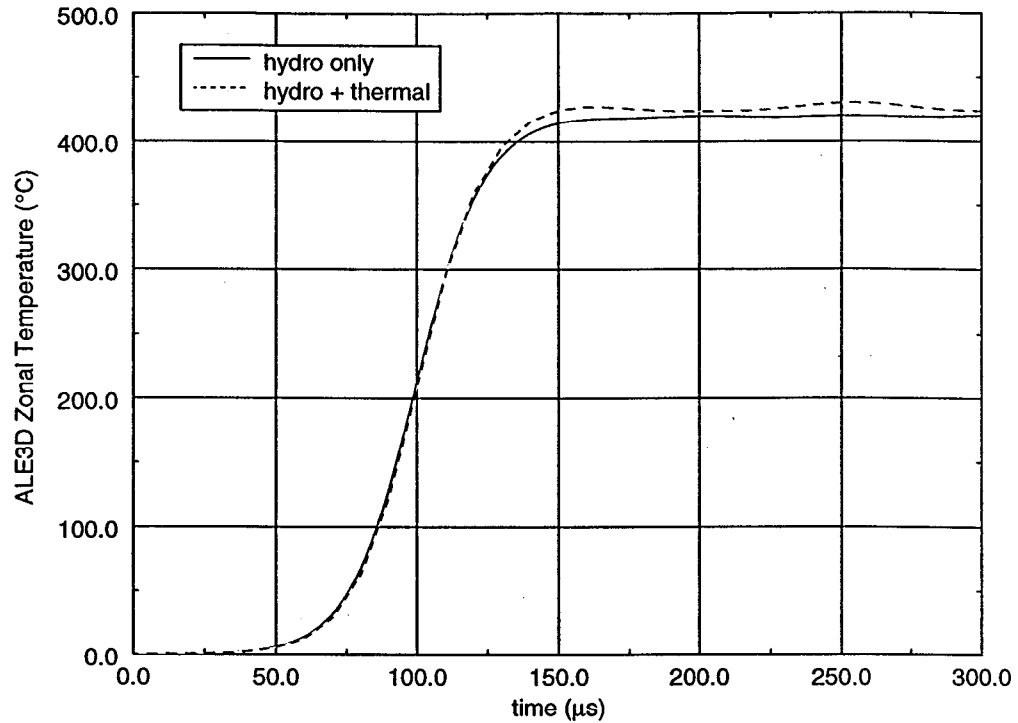


Figure 9. ALE3D zonal temperature (zone 1) for 41 μ s pulse; 'hydro only' (specified zonal energy) vs. 'hydro + thermal' (specified nodal temperature).

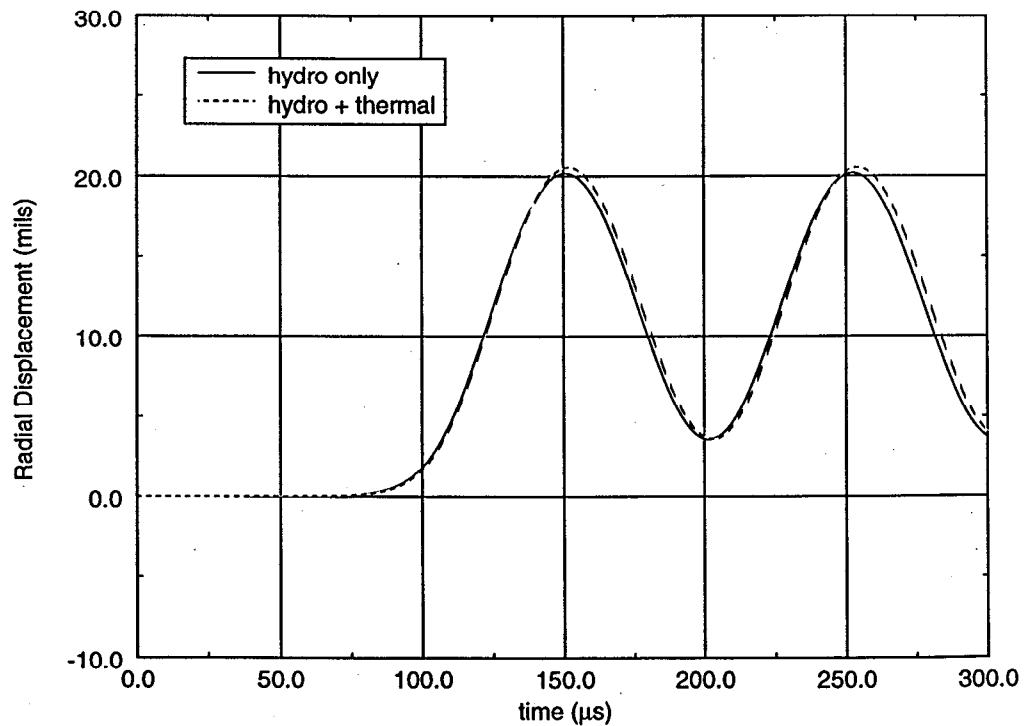


Figure 10. ALE3D inner surface displacement for 41 μ s pulse; 'hydro only' (specified zonal energy) vs. 'hydro + thermal' (specified nodal temperature).

slightly out of phase as well. The exact numerical causes for the above discrepancies are still being investigated at the time of this writing. The remaining pulse transients are calculated using the 'hydro only' approach in ALE3D.

The last issue to be examined with regard to the first pulse is that of mesh independence. To this end, the nominal mesh of Figure 6 was rezoned such that the number of elements through the thickness of the sphere was doubled. The rezoned mesh is shown in Figure 11 and has a total of 129,024 elements, with 32 elements or zones through the radial thickness. The calculations for the $41\mu\text{s}$ pulse were repeated using the 'fine' mesh in conjunction with DYNA3D and ALE3D. Table 3 compares the calculated maxima/minima boundary displacements and stresses for each code using the nominal vs. the 'fine' mesh.

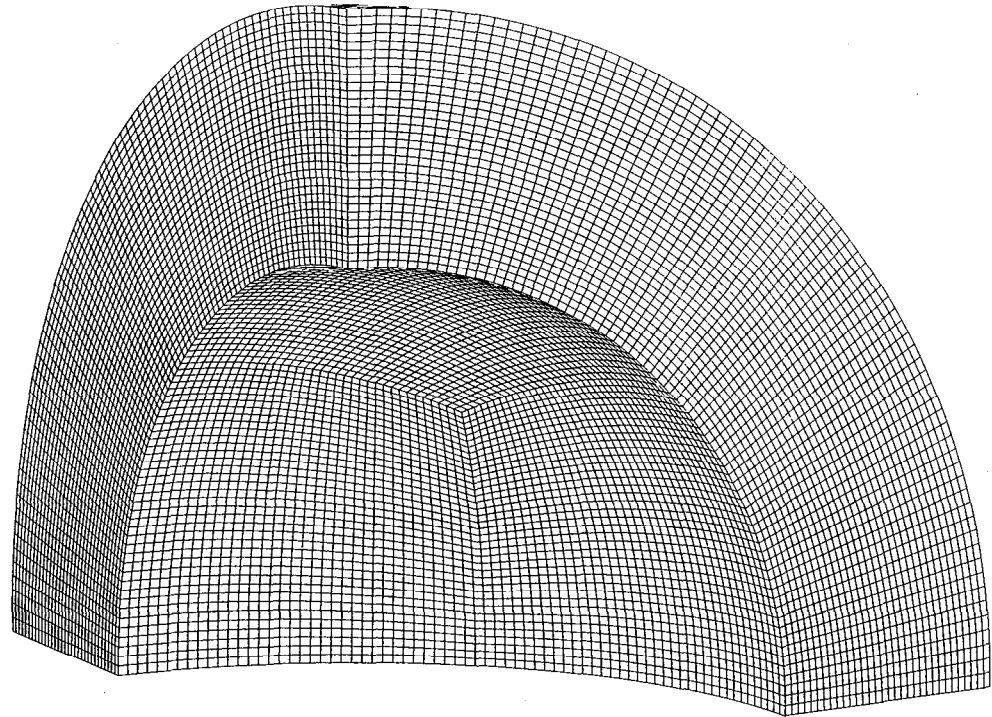


Figure 11. 'Fine' resolution finite element mesh with 32 radial elements.

TABLE 3. Comparison of Nominal vs. 'Fine' Mesh Results for $41\mu\text{s}$ FWHM, 420°C Pulse

Code	Outer Surface				Inner Surface			
	u_{max}	u_{min}	$\sigma_{\theta\theta,\text{max}}$	$\sigma_{\theta\theta,\text{min}}$	u_{max}	u_{min}	$\sigma_{\theta\theta,\text{max}}$	$\sigma_{\theta\theta,\text{min}}$
DYNA3D, nominal	23.6mils	11.7mils	38.6ksi	-37.7ksi	20.1mils	3.6mils	76.0ksi	-74.7ksi
ALE3D, nominal	23.8	11.8	38.7	-38.0	20.2	3.6	75.7	-75.2
DYNA3D, 'fine'	23.6	11.7	38.1	-37.3	20.1	3.5	76.9	-75.7
ALE3D, 'fine'	23.8	11.8	38.2	-37.6	20.2	3.6	76.7	-75.7

The results of Table 3 indicate that for a given code, essentially no change in predicted displacements occur when the mesh is rezoned. However, there are small changes (<2%) with respect to predicted hoop stresses. Namely, the outer surface hoop stress tends to decrease as the mesh is refined and inner surface hoop stress tends to increase with mesh refinement.

This trend is consistent with the previous discussion regarding 'zone centered' vs. 'nodal' variables. The inner/outer surface displacements are associated with nodes that always lie on the material boundaries, regardless of mesh refinement. On the other hand, stress is a zone centered variable. Thus, as the mesh is refined, the center of the boundary zones moves slightly closer to the actual boundary. In the case of the inner spherical surface, this tends to slightly increase the hoop stress for the finer mesh; whereas, the converse is true for the outer surface. The conclusion here is that with regard to displacements, the nominal mesh of Figure 6 is sufficiently refined. With regard to predicted hoop stress, the nominal mesh is within a few percent of convergence. Results for the remaining transients are presented using the nominal mesh of Figure 6.

3.2.2 85 μ s FWHM, 420°C Pulse

The next case to be considered more than doubles the pulse width with respect to the previous case. Here, we consider an 85 μ s FWHM pulse that again results in a peak temperature rise of 420°C. However, the pulse amplitude is decreased to 0.51MW/g, providing the same total energy input as before. All pulses considered as well as their resulting temperature rises, are shown in Figure 12.

Figure 13 shows the dynamic displacements of the inner and outer spherical surfaces for the 85 μ s pulse. As one would expect, the natural vibrational period of the oscillations remains unchanged (103 μ s). However, the amplitudes of the dynamic displacements are significantly reduced compared to the 41 μ s pulse. Roughly, the displacement amplitudes have been reduced by a factor of four and the maxima/minima displacements vary within a couple of mils about the static displacement values. Also, the first displacement oscillation consistently has a slightly lower amplitude compared to subsequent oscillations. This is possibly due to the fact that for this transient, the heating time completely overlaps the first oscillation. As before, the agreement between the four codes is quite good.

Figure 14 depicts the corresponding dynamic hoop stresses for the inner and outer surfaces. Consistent with the decrease in the displacement amplitudes, the peak hoop stresses for the inner and outer surfaces have been reduced by roughly a factor of four with respect to the 41 μ s pulse. However, the hoop stresses for this pulse are still quite significant with the inner surface again producing the highest stress of close to 20ksi tension. The greatest stress also occurs during the first oscillation, which is likely due to the extended heating time compared to the shorter 41 μ s pulse. As with the previous pulse, the finite difference code tends to slightly overshoot the inner surface hoop stress predicted by the 3D finite element codes. Specifically, the finite difference code gives 19.5ksi at $t=350\mu$ s; whereas, DYNA3D, NIKE3D, and ALE3D respectively predict 18.5ksi, 18.6ksi, and 18.5ksi. It has been shown that as the mesh is further refined, the

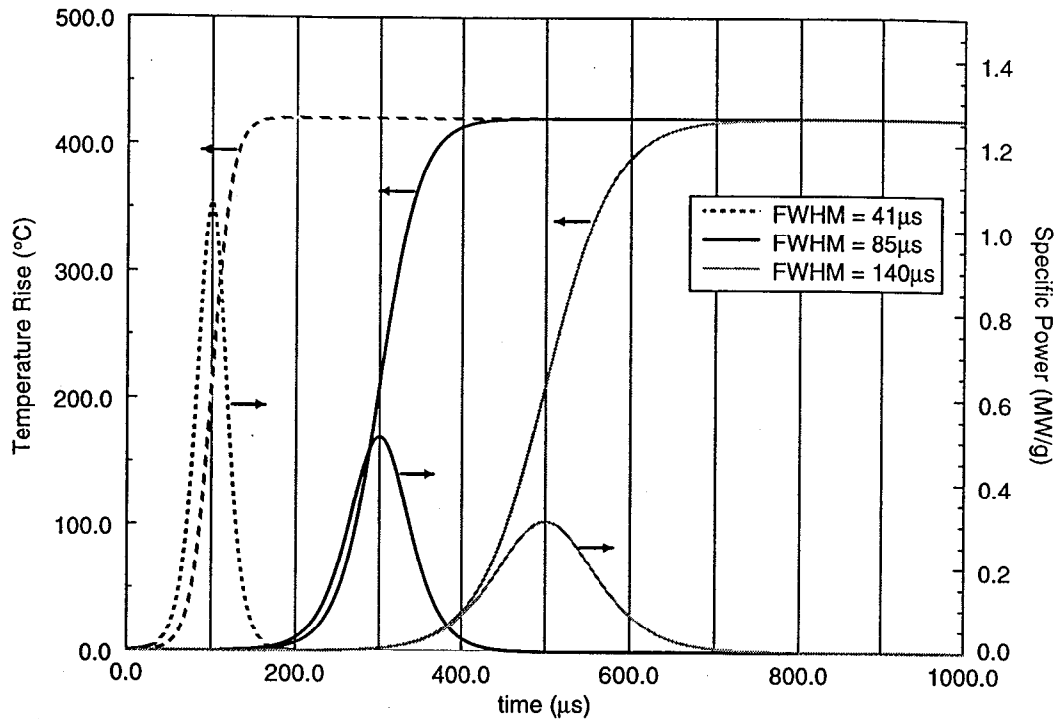


Figure 12. Temperature and specific power histories of pulses with FWHM of 41 μ s, 85 μ s, and 140 μ s.

finite element codes' stress predictions will move closer toward the finite difference results, at least as far as the inner surface is concerned.

3.2.3 140 μ s FWHM, 420°C Pulse

The last pulse to be considered is a 140 μ s FWHM pulse, shown along with its corresponding temperature rise in Figure 12. This pulse peaks at 500 μ s with a specific power of 0.31MW/g, again resulting in a 420°C rise in temperature. The predicted radial displacements are shown in Figure 15 for the boundary surfaces. This pulse is of sufficiently long duration that the displacements essentially ramp up to their static limits, or the displacement values that would result if the material were heated slowly. However, careful examination reveals that there are still oscillations at 103 μ s intervals in the calculated displacements, although the displacement amplitudes are no more than fractions of a mil. Again, the four codes are in good agreement, with all of the 3D codes resolving the subtle displacement oscillations.

Figure 16 plots the corresponding stresses for the 140 μ s pulse. Quite remarkably, despite the practically negligible displacement amplitudes, the dynamic stresses in the part are still relatively significant, a reflection of the large modulus for U-10%Mo. Again, the inner surface sets the maximum hoop stress at roughly 4000psi. The first two hoop stress oscillations are larger than subsequent oscillations as a result of the heating time extending past the second oscillation for this transient. The codes are in good agreement for this

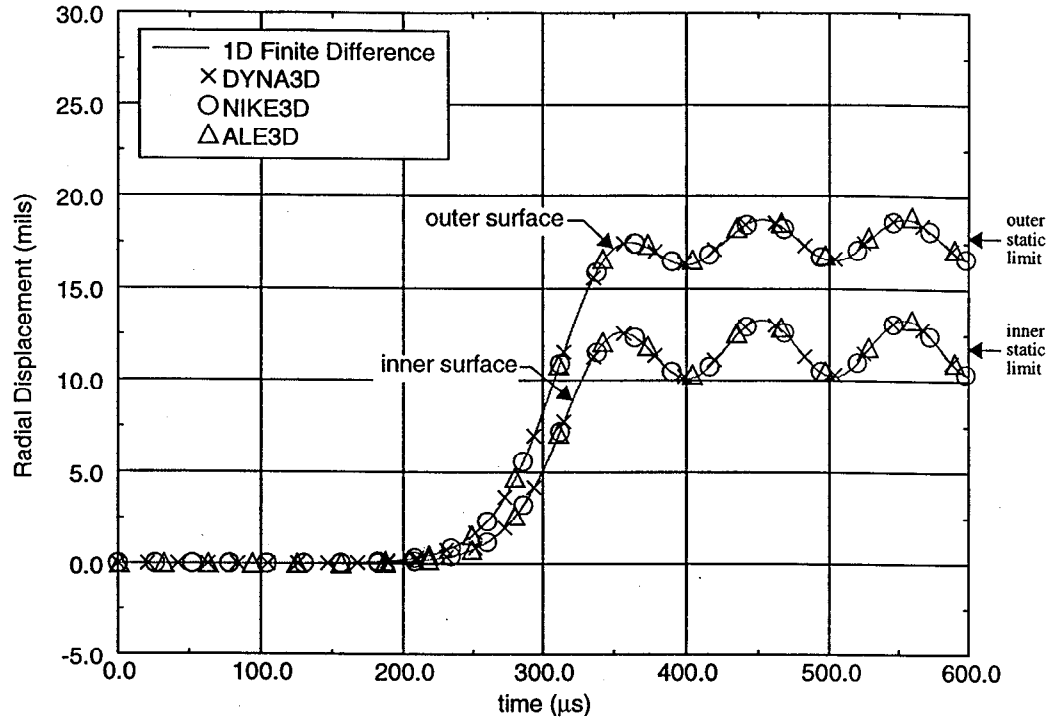


Figure 13. Code comparison of radial displacements for 85 μ s FWHM pulse with $\Delta T = 420^{\circ}\text{C}$, inner and outer surface behavior shown.

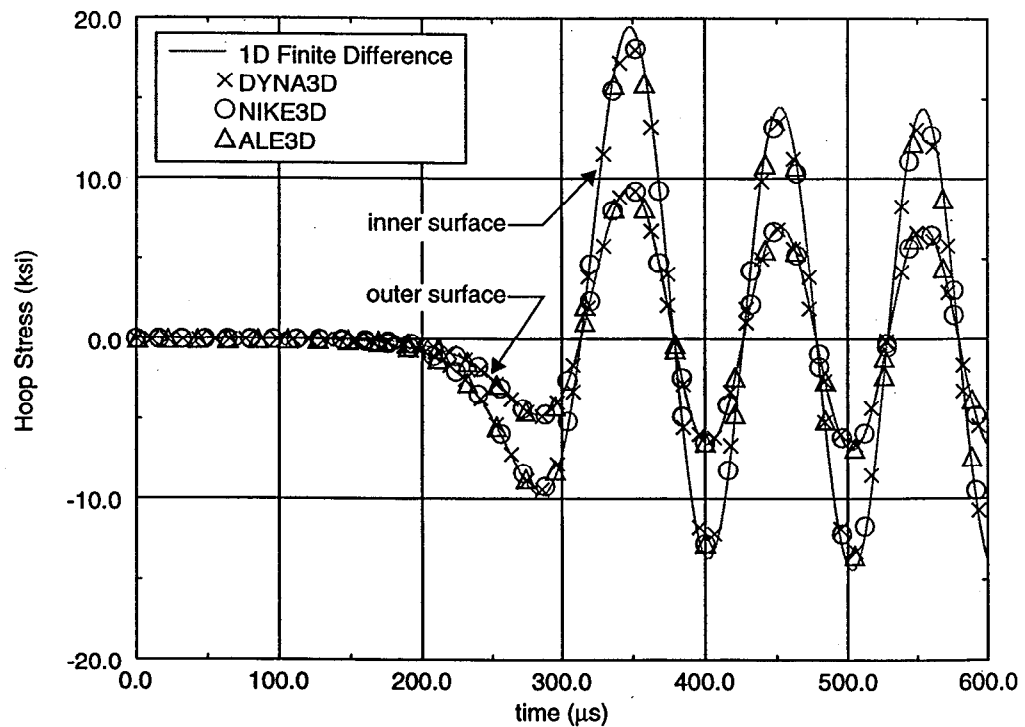


Figure 14. Code comparison of hoop stress for 85 μ s FWHM pulse with $\Delta T = 420^{\circ}\text{C}$, inner and outer surface behavior shown.

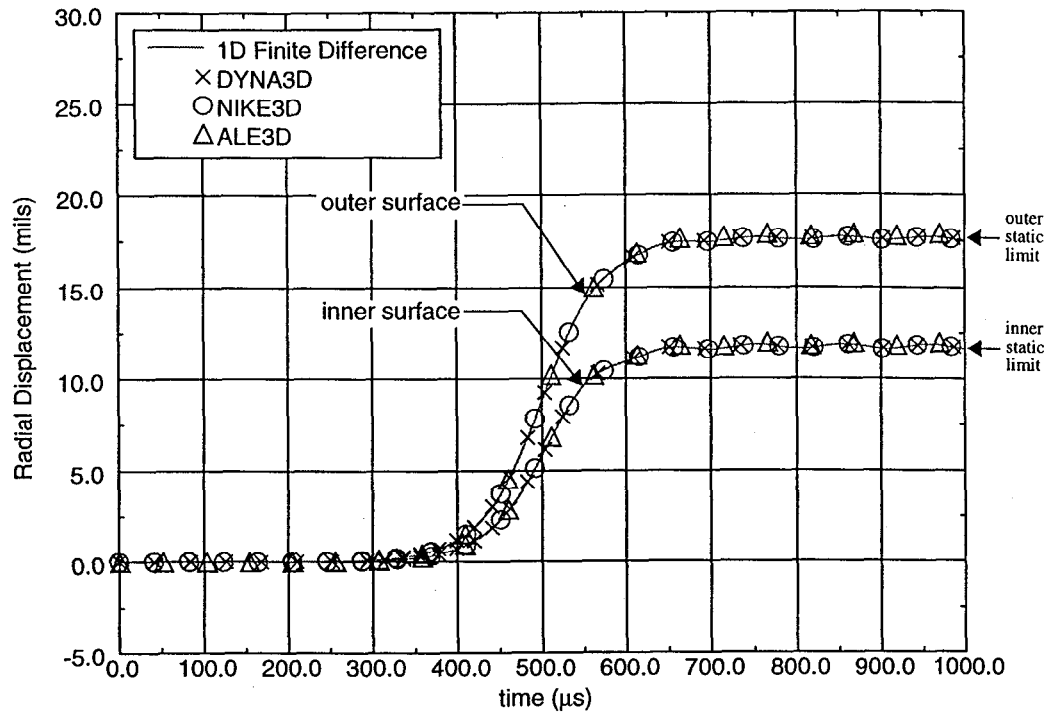


Figure 15. Code comparison of radial displacements for a 140 μ s FWHM pulse with $\Delta T = 420^{\circ}\text{C}$, inner and outer surface behavior shown.

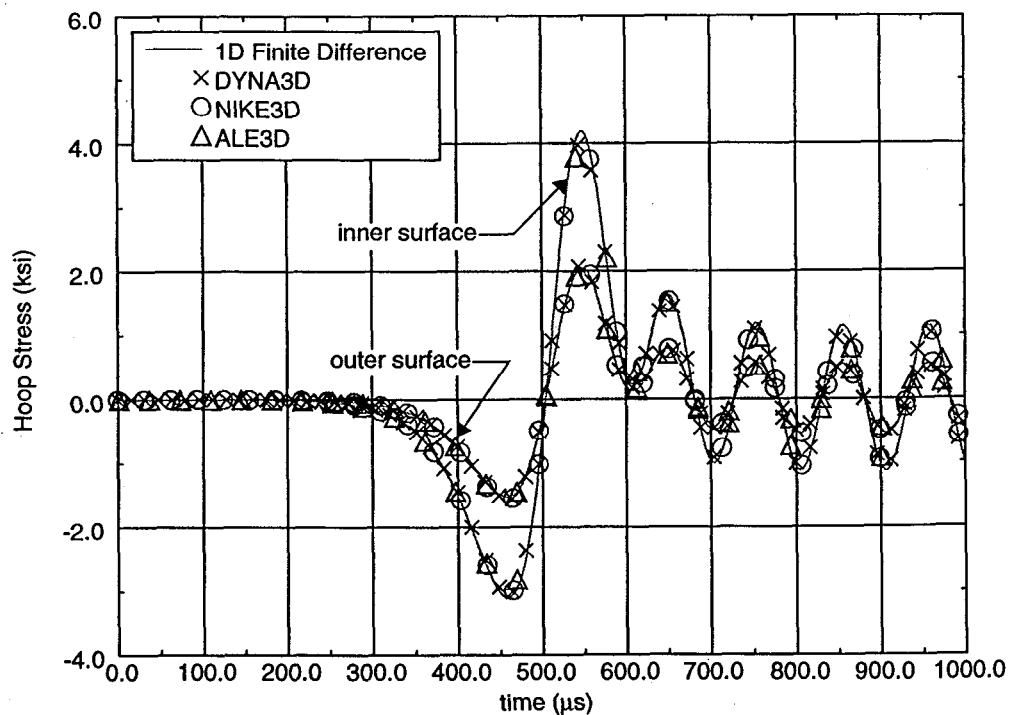


Figure 16. Code comparison of hoop stress for 140 μ s FWHM pulse with $\Delta T = 420^{\circ}\text{C}$, inner and outer surface behavior shown.

case with regard to computed stresses. There is less than a 2% deviation between code predictions for the maximum predicted hoop stress.

4.0 Conclusions and Future Work

Volumetric pulse heating of a thick-walled hollow sphere was investigated using four computer codes, each with unique capabilities. A total of three 'fission-heating' transients were considered that produced a range of dynamic effects. In general, excellent agreement was obtained in the code-to-code comparisons, which included both engineering as well as 'ASCII' codes. This process has verified the numerics of the codes for pulsed-heating dynamics in a simple geometry with simplified material models. It has also pointed out which features/options of the codes work best for these types of calculations. Of interest for future work is the application of the codes, particularly the ASCII code ALE3D, to more complicated nuclear systems for which there are accurate experimental measurements of displacement and stress. For example, the experimental data from Reuscher [5] in the Sandia Pulsed Reactor II provide an ample testbed for future modeling efforts involving more complicated material models with still relatively simple geometries.

5.0 Acknowledgments

The author would like to thank the following individuals for their generous assistance and advice with regard to this study: Richard Sharp, John Dolaghan, Rose McCallen, Al Nichols, and Andy Anderson. Also thanks to Jerry Lin and Ed Zywiec for DYNA3D support, and Mike Puso for help with NIKE3D. This work was performed under the auspices of the United States Department of Energy by Lawrence Livermore National Laboratory under contract W-7405-ENG-48.

6.0 References

- 1) J. A. Reuscher, "Thermomechanical Analysis of Fast Burst Reactors," Fast Burst Reactors - Proceedings of the National Topical Meeting on Fast Burst Reactors, pp. 51-74, CONF-690102, 1969.
- 2) R. M. Jefferson, "Sandia Pulsed Reactor II," Fast Burst Reactors - Proceedings of the National Topical Meeting on Fast Burst Reactors, pp. 105-123, CONF-690102, 1969.
- 3) D. Burgreen, "Thermoelastic Dynamics of Rods, Thin Shells, and Solid Spheres," *Nuclear Science and Engineering*, Vol. 12, pp. 203-217, 1962.
- 4) A. L. Austin, "Thermally Induced Vibrations of Concentric Thin Spherical Shells," *Nuclear Science and Engineering*, Vol. 20, pp. 45-52, 1964.
- 5) J. A. Reuscher, "Analysis of Internal Heating Shock Effects in Reactor Fuel Components," *Nuclear Engineering and Design*, Vol. 18, pp. 213-251, 1972.

- 6) G. DiPeso, "Analytic and Experimental Validation of Thermo-Elastic-Plastic Material Response Calculations," Lawrence Livermore National Laboratory, UCRL-JC-128704, October 2, 1997.
- 7) R. G. Whirley & B. E. Engelmann, "DYNA3D: A Nonlinear, Explicit, Three-Dimensional Finite Element Code For Solid and Structural Mechanics - User Manual," Lawrence Livermore National Laboratory, UCRL-MA-107254, 1993.
- 8) G. DiPeso, "Thermoelasticplastic Response of Thin Spherical Shells," Lawrence Livermore National Laboratory, QDV-97-0004, 1997.
- 9) B. N. Maker, "NIKE3D: A Nonlinear, Implicit, Three-Dimensional Finite Element Code for Solid and Structural Mechanics - User Manual," Lawrence Livermore National Laboratory, UCRL-MA-105258, 1995.
- 10) E. Dube, R. Neely, A. Nichols, R. Sharp, and R. Couch, "User's Manual for ALE3D, An Arbitrary Lagrange/Eulerian 3D Code System," Version 2.8.5, Lawrence Livermore National Laboratory, B-Division, October 14, 1999.
- 11) J. J. Duderstadt & L. J. Hamilton, Nuclear Reactor Analysis, pp. 45-50, John Wiley and Sons, New York, 1976.
- 12) J. A. Reuscher, personal communication, Texas A&M University, 1990.
- 13) S. Timoshenko & J. N. Goodier, Theory of Elasticity, pp 452-456, McGraw-Hill, New York, 1951.
- 14) T. F. Wimett, R. H. White, W. R. Stratton, and D. P. Wood, "Godiva II - An Unmoderated Pulse-Irradiation Reactor," *Nuclear Science and Engineering*, Vol. 8, pp. 691-708, 1960.
- 15) TRUEGRID, XYZ Scientific Applications, Inc., 1324 Concannon Blvd., Livermore, CA.

7.0 APPENDIX A

This appendix contains the code listing for the 1D finite-difference code 'newpulse.f', as described in Section 2.2.1. Many variables are set during interactive execution; however, others are not and thus require code editing and recompilation. For example, the desired spatial temperature function must be defined in the code as well as the desired material properties. The default material is U-10%Mo with constant properties.

```

      program newpulse
c
c
c      code variables
c
c      material property variables, input in cgs units
c
c      yngmod      = Young's Modulus (dynes/cm^2)
c      csonic      = speed of sound in fuel (cm/s)
c      density      = fuel density (g/cm^3)
c      cte          = th. expansion coeff. of fuel (1/C)
c      poiss        = Poisson's Ratio
c
c      other variables created during code execution
c
c      disp1(i)    = displacement of node i at time k-1, cm
c      disp2(i)    = displacement of node i at time k, cm
c      disp3(i)    = displacement of node i at time k+1, cm
c      del         = distance between nodal mesh points, cm
c      deltat      = the time step, s
c      maxtime     = maximum number of time steps to be taken
c      nodes       = number of nodes
c      npmax       = maximum number of nodes
c      nprt        = print interval
c      rinner      = inner reactor radius, cm
c      router      = outer reactor radius, cm
c      strain1     = inner node radial strain
c      strainn     = outer node radial strain
c      time        = total elapsed time since beginning the transient, s
c      tpo2        = time to peak power, s
c      temp        = temporal temperature function at time k+1, C
c      tmax        = maximum temperature reached, C
c      tau1        = hoop stress at inner node, dynes/cm^2
c      taun        = hoop stress at outer node, dynes/cm^2
c      func(i)     = spatial temperature profile, fixed for short timescales
c      fwhm        = pulse width, s
c      x(i)        = radial distance from reactor CL to node i, cm
c
c      *****
c      define required data and control parameters
c      *****
c
c      implicit double precision(a-h,o-z), integer(i-n)

```

```

parameter(npmax=10000)
c
c
dimension x(npmax),func(npmax),disp1(npmax)
dimension disp2(npmax),disp3(npmax)
c
c OUTPUT FILES AND UNITS
c ***NOTE, output units differ from input units***
c time plot, inner surface temperature, us vs. C
open(unit=11, file='temp1.out',status='new')
c time plot, outer surface temperature, us vs. C
open(unit=12, file='tempn.out',status='new')
c time plot, inner surface radial displacement, us vs. mils
open(unit=13, file='disp1.out',status='new')
c time plot outer surface radial displacement, us vs. mils
open(unit=14, file='dispn.out',status='new')
c time plot inner surface hoop stress, us vs. ksi
open(unit=15, file='taul.out',status='new')
c time plot outer surface hoop stress, us vs. ksi
open(unit=16, file='taun.out',status='new')
c spatial temperature profile, inches vs C
open(unit=17, file='profile.out',status='new')
c inner surface hoop stress vs. strain, in/in vs. ksi
open(unit=18, file='svsel.out',status='new')
c outer surface hoop stress vs. strain, in/in vs. ksi
open(unit=19, file='svse2.out',status='new')
c
c ***define the geometry here***
c g=1 for a cylinder, 2 for a sphere
c
write(*,*)
write(*,*) 'NEWPULSE, R.E. Canaan based on J.A. Reuscher, 1999'
write(*,*)
write(*,*) 'Enter 1 for a hollow cylinder, 2 for a hollow sphere'
read(*,*) g
c
write(*,*)
c
if(g.eq.1) then
write(*,*) '***Cylindrical Geometry***'
endif
if(g.eq.2) then
write(*,*) '***Spherical Geometry***'
endif
c
write(*,*)
c
write(*,*) 'Enter the inner radius in cm'
read(*,*) rinner
c
write(*,*) 'Enter the outer radius in cm'
read(*,*) router
c
write(*,*)

```

```

write(*,*) '***NOTE:***'
write(*,*) 'The time step and spatial discretization are'
write(*,*) 'specified internally and must meet the appropriate'
write(*,*) 'stability criteria!'
write(*,*)

pi=4.0*atan(1.0)

c
c
c ***define the time step, it must meet the stability criteria***
c
c [time duration of transient=deltat*maxtime]
c deltat = 1.0e-08
c write(*,*) 'Using a time step of ',deltat*1.0e+06,' us.'
c write(*,*)
c maxtime = 30000
c write(*,*) 'Run duration = ',maxtime*deltat*1.0e+06,' us.'
c write(*,*)

c
c ***define the spatial discretization***
c
c nodes = 100
c write(*,*) 'Using a total of ',nodes,' node points'
c write(*,*)

c
c ***these constants determine the temperature pulse shape using
c Reusher's burst function***
c
c write(*,*) 'SPECIFY PULSE CHARACTERISTICS'
c write(*,*)
c write(*,*) 'Enter the FWHM of the pulse in us'
c read(*,*) fwhm
c fwhm=fwhm*1.0e-06

c
c write(*,*) 'Enter the time to peak power in us'
c read(*,*) tpo2
c tpo2=tpo2*1.0e-06

c
c write(*,*) 'Enter the desired max. temperature rise, in C'
c read(*,*) tmax

c
c write(*,*)

c
c ***print output to file every npert timesteps***
c
c npert = 100

c
c ***define material properties in cgs units***
c
c data yngmod,poiss,density,cte/9.0e+11,0.3,17.2,1.4e-05/

c
c write(*,*) 'Material properties based on U-10%Mo'
c write(*,*)

c
c determine the speed of sound, c, in cm/s

```

```

cnum=yngmod*(1.0-poiss)
cden=(1.0+poiss)*(1.0-2.*poiss)*density
csonic=(cnum/cden)**0.5
C
C
C *****
C   set up the mesh
C *****
C
C   del=(router-rinner)/(nodes-1)
C
C   ***define the spatial temperature distribution***
C
C   write(*,*) 'Select the desired spatial temperature profile:'
C   write(*,*)
C   write(*,*) '***WARNING***'
C   write(*,*) 'Selections 2 and 3 should be tailored for a'
C   write(*,*) 'specific geometry. Please edit the source code'
C   write(*,*) 'and specify a spatial temperature profile which'
C   write(*,*) 'is correct for your geometry'
C   write(*,*)
C   write(*,*) ' Enter 1 for uniform temperature'
C   write(*,*) ' Enter 2 for an arbitrary cosine distribution'
C   write(*,*) ' Enter 3 for an arbitrary fission spectrum'
C   read(*,*) iflag
C
C   write(*,*)
C   write(*,*) 'INPUT COMPLETE'
C   write(*,*)
C   write(*,*) 'FOR OUTPUT UNITS, SEE CODE COMMENTS'
C
C   x(1)=rinner
C
C   if(iflag.eq.1) then
C     func(1) = 1.0
C   elseif(iflag.eq.2) then
C     func(1)=cos((pi/20.32)*(x(1)-2.0))
C   else
C     func(1)=0.41482+(0.35594*x(1))-(0.064311*x(1)**2) +
&      (0.0027732*x(1)**3)
C   endif
C
C   x(nodes)=router
C
C   if(iflag.eq.1) then
C     func(nodes) = 1.0
C   elseif(iflag.eq.2) then
C     func(nodes)=cos((pi/20.32)*(x(nodes)-2.0))
C   else
C     func(nodes)=0.41482+(0.35594*x(nodes)) -
&      (0.064311*x(nodes)**2)+(0.0027732*x(nodes)**3)
C   endif
C
C   do 10 i=2,nodes-1

```

```

        x(i)=x(i-1) + del
c
        if(iflag.eq.1) then
            func(i) = 1.0
        elseif(iflag.eq.2) then
            func(i)=cos((pi/20.32)*(x(i)-2.0))
        else
            func(i)=0.41482+(0.35594*x(i))-(0.064311*x(i)**2) +
&            (0.0027732*x(i)**3)
            endif
10 continue
c
c *****
c     apply the initial conditions
c *****
c
        do 20 i=1,nodes
            disp1(i)=0.0
            disp2(i)=0.0
            disp3(i)=0.0
20 continue
c
c *****
c     define the coefficients for the displacement equs.
c *****
c
        termal=(csonic*csonic*deltat*deltat)/(del*del)
        terma2=(g*csonic*csonic*deltat*deltat)/(2.0*del)
        termb1=2*termal
        termb2=g*csonic*csonic*deltat*deltat
        termc1=termal
        termc2=terma2
        termd1=((1+poiss)/(1-poiss))*(cte*terma2/g)
c
c     ***calculate other needed constants***
c
        xlam=(poiss*yngmod)/((1.0+poiss)*(1.0-2.0*poiss))
        xmu=yngmod/(2.0*(1.0+poiss))
c
c *****
c     begin the time march
c *****
c
        do 100 k=1,maxtime
            time=k*deltat
            temp=tmax/(exp((-3.52/fwhm)*(time-tpo2)) + 1.0)
c
c     ***solve for displacements in nodes 2 thru n-1***
c
            do 35 i=2,nodes-1
                disp3(i)=disp2(i+1)*(termal+terma2/x(i)) +
&                disp2(i)*(2.0-termb1-termb2/x(i)**2) +
&                disp2(i-1)*(termc1-termc2/x(i)) -
&                termd1*(temp*func(i+1) - temp*func(i-1)) -

```

```

      &      disp1(i)
35      continue
c
c      ***solve for displacements at the boundary nodes***
c
c      inner node
c
c          terme1=3.0*xlam + 2.0*xmu
c          terme2=2.0*xmu + xlam
c
c          xnum1=disp3(3)-4.0*disp3(2)+(terme1/terme2)*
&          2.0*del*cte*temp*func(1)
c          if(g.eq.1) then
c              xden1=(2.0*del*xlam)/(terme2*x(1)) - 3.0
c          endif
c
c          if(g.eq.2) then
c              xden1=(2.0*2.0*del*xlam)/(terme2*x(1)) - 3.0
c          endif
c          disp3(1)=xnum1/xden1
c
c      inner node strain
c
c      strain1=disp3(1)/rinner
c
c      outer node
c
c          xnum2=4.0*disp3(nodes)-disp3(nodes-2) +
&          (terme1/terme2)*2.0*del*cte*temp*func(nodes)
c          if(g.eq.1) then
c              xden2=(2.0*del*xlam)/(terme2*x(nodes)) + 3.0
c          endif
c
c          if(g.eq.2) then
c              xden2=(2.0*2.0*del*xlam)/(terme2*x(nodes)) + 3.0
c          endif
c          disp3(nodes)=xnum2/xden2
c
c      outer node strain
c
c      strainn=disp3(nodes)/router
c
c      ***calculate the hoop stress at inner node, node 1***
c
c          if(g.eq.1) then
c              tau1=(xlam/(2.0*del))*(-3.0*disp3(1) + 4.0*disp3(2) -
&              disp3(3)) + (terme2*disp3(1))/x(1) -
&              terme1*cte*temp*func(1)
c          endif
c
c          if(g.eq.2) then
c              tau1=(xlam/(2.0*del))*(-3.0*disp3(1) + 4.0*disp3(2) -
&              disp3(3)) + (terme2*disp3(1))/(x(1)*(1.0-pois)) -
&              terme1*cte*temp*func(1)

```



```

endif
c
c   ***calculate the hoop stress at outer node***
c
      if(g.eq.1) then
        taun=(xlam/(2.0*del))*(3.0*disp3(nodes) -
&      4.0*disp3(nodes-1) + disp3(nodes-2)) +
&      (terme2*disp3(nodes))/x(nodes) -
&      termel*cte*temp*func(nodes)
      endif
c
      if(g.eq.2) then
        taun=(xlam/(2.0*del))*(3.0*disp3(nodes) -
&      4.0*disp3(nodes-1) + disp3(nodes-2)) +
&      (terme2*disp3(nodes))/(x(nodes)*(1.0-pois)) -
&      termel*cte*temp*func(nodes)
      endif
c
c   print out results every nprt timesteps
c
c   ***convert time to microsecond, displacement to mils
c   and stress to ksi***
c
      if(mod(k,nprt).eq.0) then
        write(11,110) time*1.0e+06, temp*func(1)
        write(12,120) time*1.0e+06, temp*func(nodes)
        write(13,130) time*1.0e+06, disp3(1)*3.937008e+02
        write(14,140) time*1.0e+06, disp3(nodes)*3.937008e+02
        write(15,150) time*1.0e+06, tau1*1.450378e-08
        write(16,160) time*1.0e+06, taun*1.450378e-08
        write(18,180) strain1, tau1*1.450378e-08
        write(19,190) strainn, taun*1.450378e-08
      endif
c
c   ***print out spatial temp. dist. at last time step***
c
      if(k.eq.maxtime) then
        do 40 i=1,nodes
          write(17,170) x(i)/2.54, temp*func(i)
40      continue
        endif
c
c   ***update the displacements for the next timestep***
c
      do 45 i=1,nodes
        disp1(i)=disp2(i)
        disp2(i)=disp3(i)
45      continue
c
c   ***proceed to next time step***
c
100 continue
c
c

```

```
c
c
110 format(' ',g12.5,5x,g12.5)
120 format(' ',g12.5,5x,g12.5)
130 format(' ',g12.5,5x,g12.5)
140 format(' ',g12.5,5x,g12.5)
150 format(' ',g12.5,5x,g12.5)
160 format(' ',g12.5,5x,g12.5)
170 format(' ',g12.5,5x,g12.5)
180 format(' ',g12.5,5x,g12.5)
190 format(' ',g12.5,5x,g12.5)
```

```
c
c
c
write(*,*)
write(*,*) 'DONE'
stop
end
```

8.0 APPENDIX B

The following is the DYNA3D input deck (excluding node and element definitions) for the first pulse transient considered, the 41 μ s FWHM pulse with a maximum temperature rise of 420°C. The general approach is described in Section 2.2.2.

```
dyna hollow sphere                                     88 large
*
* This file was created using TrueGrid by XYZ Scientific Applications, Inc.
* For further information, call (510) 373-0628 or write to:
*
*     XYZ Scientific Applications, Inc.
*     1324 Concannon Blvd.
*     Livermore, Ca. 94550
*
*
*----- ANALYSIS INPUT DATA FOR DYNA3D -----*
* TrueGrid version 1.3.23 dated 12/15/97
* generated on Sep  3 1999 at 15:26:00
*
*----- CONTROL CARD #2 -----*
*
* number of materials[1] nodal points[2] solid hexahedron elements[3] beam
* elements[4] 4-node shell elements[5] 8-node solid shell elements[6]
* interface segments[7] interface interval[8] min. shell time step[9]
*     1      22831      20480      0      0      0      0 0.000E+00  0.0
*
*----- CONTROL CARD #3 -----*
*
* number of time history blocks for nodes[1] hexahedron elements[2] beam
* elements[3] shell elements[4] thick shell elements[5] and report interval[6]
* reaction forces print flag[7] discrete element forces print flag[8]
* element deletion/SAND database flag[9]
*     0      0      0      0      0      0      0      0      1
*
*----- CONTROL CARD #4 -----*
*
* number of sliding boundary planes[1]
* sliding boundary planes w/ failure[2] points in density vs depth
* curve[3] brode function flag[4] number of rigid body merge cards[5]
* nodal coordinate format[6] force cross sections[7]
* cross section forces interval[8]
*     0      0      0      0      0e20.9      0 0.000E+00
*
*----- CONTROL CARD #5 -----*
*
* number of load curves[1] concentrated nodal loads[2] element sides having
* pressure loads applied[3] velocity/acceleration boundary condition cards[4]
* rigid walls (stonewalls)[5] nodal constraint cards[6] initial condition
* parameter[7] sliding interfaces[8] base acceleration in x[9] y[10] and
* z-direction[11] angular velocity about x[12] y[13] and z-axis[14] number of
* solid hexahedron elements for momentum deposit[15] detonation points[16]
*     1      0      0      0      0      0      0      0      0      0      0      0      0      0      0
*
*----- CONTROL CARD #6 -----*
*
* termination time[1] time history dump interval[2] complete dump interval[3]
* time steps between restart dumps[4] time steps between running restart
```

```

* dumps[5] initial time step[6] sliding interface penalty factor[7] thermal
* effects option[8] default viscosity flag[9] computed time step factor[10]
3.000E-04 0.000E+00 1.000E-06 0 100 0.000E+00 0.000E+00 1 0 0.000E+00
*
*----- CONTROL CARD #7 -----*
*
* number of joint definitions[1] rigid bodies with extra nodes[2] shell-
* solid interfaces[3] tie-breaking shell slidelines[4] tied node sets with
* failure[5] limiting time step load curve number[6] springs-dampers-masses
* flag[7] rigid bodies with inertial properties[8] dump shell strain flag[9]
* Hughes-Liu shell update[10] shell thickness change[11]
* shell formulation[12] number of nonreflecting boundary segments[13]
0 0 0 0 0 0 0 0 0 0 0 0 0
*
*----- CONTROL CARD #8 -----*
*
* number of point constraint nodes[1] coordinate systems for constraint
* nodes[2] minimum step factor[3] number of beam integration rules[4]
* maximum integration points for beams[5] number of shell integration rules[6]
* maximum integration points for shells[7] relaxation iterations between
* checks[8] relaxation tolerance[9] dynamic relaxation factor[10] dynamic
* relaxation time step factor[11] 4-node shell time step option[12]
0 0 0.000E+00 0 0 0 0 250 1.000E-03 9.950E-01 0.000E+00 0
*
*----- CONTROL CARD #9 -----*
*
* plane stress plasticity[1] printout flag[2] number of 1D slidelines[3]
* relaxation database[4] Rayleigh coefficient[5] materials w/Rayleigh damping[6]
0 0 0 0 0.000E+00 0
*
*----- MATERIAL CARDS -----*
*
1 41.7200E+01 0 00.0000E+00 00.0000E+00 0.0000E+00 0 0 0
U-10Mo
0.000E+00 5.000E+02 0.000E+00 0.000E+00 0.000E+00 0.000E+00 0.000E+00 0.000E+00
9.000E+11 9.000E+11 0.000E+00 0.000E+00 0.000E+00 0.000E+00 0.000E+00 0.000E+00
3.000E-01 3.000E-01 0.000E+00 0.000E+00 0.000E+00 0.000E+00 0.000E+00 0.000E+00
1.400E-05 1.400E-05 0.000E+00 0.000E+00 0.000E+00 0.000E+00 0.000E+00 0.000E+00
0.000E+00 0.000E+00 0.000E+00 0.000E+00 0.000E+00 0.000E+00 0.000E+00 0.000E+00
0.000E+00 0.000E+00 0.000E+00 0.000E+00 0.000E+00 0.000E+00 0.000E+00 0.000E+00
*
*----- NODE DEFINITIONS -----*
*
*----- ELEMENT DEFINITIONS -----*
*
*----- LOAD CURVES -----*
*
1 23 0
0.000E+00 0.000E+00
1.000E-05 1.851E-01
2.000E-05 4.365E-01
3.000E-05 1.028E+00
4.000E-05 2.419E+00
5.000E-05 5.663E+00
6.000E-05 1.312E+01
7.000E-05 2.970E+01
8.000E-05 6.394E+01
9.000E-05 1.250E+02
1.000E-04 2.100E+02

```

1.100E-04	2.950E+02
1.200E-04	3.561E+02
1.300E-04	3.903E+02
1.400E-04	4.069E+02
1.500E-04	4.143E+02
1.600E-04	4.176E+02
1.700E-04	4.190E+02
1.800E-04	4.196E+02
1.900E-04	4.198E+02
2.000E-04	4.199E+02
2.100E-04	4.200E+02
3.000E-04	4.200E+02

10.0 APPENDIX D

The following are ALE3D generator input files for the first pulse transient considered, the 41 μ s FWHM pulse with a maximum temperature rise of 420°C. Two generator input files are included in this appendix. The first, directly below, uses the 'heatgen fixed' keyword in the Boundary Block in order to specify the fission pulse in terms of specific power vs. time according to Equation 10. Thus, only the Hydro Block is required.

The second file specifies the pulse in terms of nodal temperature vs. time according to Equation 9. This latter approach uses the 'heat temperature' keyword and therefore requires fully coupled hydro/thermal diffusion with both a Hydro and Thermal Block. For details, see the discussion in Section 2.2.4.

ALE-GENERATOR INPUT FILE 1

```
# ale hollow sphere
#
# unit system is B Div (g,cm,micros,Mbar,Mbar-cc)
#
OUTPUT

dumptime 100.0 300.0      #dump restart files every 100microsecs
dumpac 1                  #exception - dump at cycle 1
restartvar add x y z x0 y0 z0 #add these to the defaults on p.39
plotvar add x y z x0 y0 z0 xdiff ydiff zdiff sx sy sxnorm synorm

plottime 100.0 300.0      #plot file every 100microsecs

# Create derived variables representing the radial displacements
# of nodes with respect to their original positions.

derivedvar xdiff diff x x0 #variable xdiff=x-x0
derivedvar ydiff diff y y0 #variable ydiff=y-y0
derivedvar zdiff diff z z0 #variable zdiff=z-z0

# Create variables to convert deviatoric stresses (sx,sy)
# to normal stresses (sxnorm,synorm).

derivedvar sxnorm diff sx p #variable sxnorm=sx-p
derivedvar synorm diff sy p #variable synorm=sy-p

# Collect time histories of displacement and stress for
# interior and exterior nodes that lie on the x,y,& z axes.
# These nodes have been constrained to only move radially.

timehist x 15352 22392
timehist y 15352 22392 #as a check on symmetry bc
timehist z 15352 22392 #as a check on symmetry bc

timehist y 8212 8532
timehist x 8212 8532 #as a check on symmetry bc
timehist z 8212 8532 #as a check on symmetry bc

timehist z 1 17
timehist x 1 17 #as a check on symmetry bc
timehist y 1 17 #as a check on symmetry bc
```

```

timehist xdiff 15352 22392
timehist ydiff 8212 8532
timehist zdiff 1 17

timehist sxnorm 1 6601
timehist sxnorm 7041 7341
timehist synorm 13441 13456

timehist tkelv 1 6601 7041 7341 13441 13456 #zonal temperature
timehist nodet 1 17 #nodal temperature, only works with thermal block

timehist p 1 6601 7041 7341 13441 13456 #zonal pressure
timehist e 1 6601 #zonal energy
timehist v 1 6601 #relative volume

timehist_time 1.0 300.    #plot time histories every microsecond
stoptime 300.            #simulation ends at 300microseconds

notify 1

END

CONTROL

#energy,pressure,velocity,stress cutoffs
e_cut 1.0e-12
p_cut 1.0e-12
u_cut 1.0e-12
strsscut 1.0e-15

#heat gen. rate per unit mass, variation for material 1 w/time
#41microsec FWHM, 420C max temp increase
table 1 noop #units are time in microsecs vs q in Mbar-cc/g-microsec
0.0 7.8948E-09
10.0 1.8619E-08
20.0 4.3884E-08
30.0 1.0326E-07
40.0 2.4205E-07
50.0 5.6233E-07
60.0 1.2796E-06
70.0 2.7783E-06
80.0 5.4562E-06
90.0 8.8374E-06
100.0 1.0568E-05
110.0 8.8374E-06
120.0 5.4562E-06
130.0 2.7783E-06
140.0 1.2796E-06
150.0 5.6233E-07
160.0 2.4205E-07
170.0 1.0326E-07
180.0 4.3884E-08
190.0 1.8619E-08
200.0 7.8948E-09
210.0 3.3463E-09
220.0 1.4183E-09

```

```

230.0 6.0105E-10
240.0 2.5472E-10
250.0 1.0794E-10
260.0 4.5745E-11
270.0 1.9386E-11
280.0 8.2157E-12
290.0 3.4816E-12
300.0 1.4754E-12
301.0 1.3540E-12

```

```

#initial timestep 0.01 microseconds
dtinit 0.01

```

END

HYDRO

```

linqflag 1      #use speed of sound for linear art. viscosity

#use defaults: explicit hydro, purely LaGrangian

#initialize engr material
efrmtp 1        #initialize energies and volumes based on input-
#               pressures and temperatures.
efrmtptol 1.0e-8

```

END

```

MATERIAL u-10mo 1
matinput
t0 0.1 e0 1.e-10 v0 1.0 p0 0.0 rho 17.2
czero 1.4 qfb 0.0 crq 0.1 pmin -1.0
enginput isol 54 #engr. model 54, akin to dyna3d model 4
initialyieldstress #units in Mbar, set high to never yield
0.1 100. 100. 100. 200. 100. 300. 100. 400. 100. 500. 100.
youngsmo        #units in Mbar
0.1 0.9 100. 0.9 200. 0.9 300. 0.9 400. 0.9 500. 0.9
poissonratio #unitless
0.1 0.3 100. 0.3 200. 0.3 300. 0.3 400. 0.3 500. 0.3
coeffthermalexpansion #volumetric_cte=3.0*linear_cte
0.1 4.2e-5 100. 4.2e-5 200. 4.2e-5 300. 4.2e-5 400. 4.2e-5
500. 4.2e-5
plasticmod
0.1 100. 100. 100. 200. 100. 300. 100. 400. 100. 500. 100.
tblcv #directly enter values for cv, in Mbar-cc/gC
0.1 1.17236e-06 100. 1.17236e-06 200. 1.17236e-06
300. 1.17236e-06 400. 1.17236e-06 500. 1.17236e-06

```

END

THERMAL

```

heatcond off      #no thermal, ie no heat conduction

```

END